Annals of Pure and Applied Logic 103 (2000) 155–199

ANNALS OF PURE AND APPLIED LOGIC

A bounded arithmetic *AID* for Frege systems

Toshiyasu Arai *

Faculty of Integrated Arts and Sciences, Hiroshima University, Higashi-Hiroshima, 739-8521 Japan

Communicated by Ph.G. Kolaitis

Abstract

In this paper we introduce a system *AID* (alogtime inductive definitions) of bounded arithmetic. The main feature of *AID* is to allow a form of inductive definitions, which was extracted from Buss' propositional consistency proof of Frege systems $\mathscr{F}$ in Buss (Ann. Pure Appl. Logic 52 (1991) 3–29). We show that *AID* proves the soundness of $\mathscr{F}$, and conversely any Σ^{b}_0-theorem in *AID* yields boolean sentences of which $\mathscr{F}$ has polysize proofs. Further we define Σ^{b}_1-faithful interpretations between $AID + \Sigma^{\mathrm{b}}_0$-CA and a quantified theory *QALV* of an equational system *ALV* in Clote (Ann. Math. Art. Intell. 6 (1992) 57–106). Hence *ALV* also proves the soundness of $\mathscr{F}$. © 2000 Elsevier Science B.V. All rights reserved.

MSC: 03F20; 03F30; 68Q15

Keywords: Frege system; *ALOGTIME*; Bounded arithmetic

There are two sources by Cook [15] and Buss [7] to motivate this paper.

In the pioneering paper [15] Cook shows that his equational theory *PV* corresponds to the *extended Frege system* $e\mathscr{F}$. This means that *PV* proves the soundness of $e\mathscr{F}$, and any provable equation in *PV* can be transformed into boolean tautologies so that $e\mathscr{F}$ has polysize proofs of these tautologies. Thus an extended Frege system has also polysize proofs of partial consistencies of itself. Later, Buss [4] shows that the same thing holds for the Σ^{b}_1-theorems of the bounded arithmetic S^1_2 in place of *PV*, and $e\mathscr{F}$. Note that in these theories *PV* and S^1_2 characterize the complexity class *P*.

It had remained open as to what bounded arithmetic $T_{\mathscr{F}}$ corresponds to the *Frege system* $\mathscr{F}$. That is, a bounded arithmetic $T_{\mathscr{F}}$ such that $T_{\mathscr{F}}$ proves the soundness of $\mathscr{F}$, and for any formula in a restricted form if $T_{\mathscr{F}}$ proves the formula, then it can be transformed into boolean sentences so that $\mathscr{F}$ has polysize proofs of these sentences. In [10] Clote defines a function algebra N_0 and shows that $N_0 = \mathscr{F}ALOGTIME$, the class of *ALOGTIME*-computable functions. Then he [11] introduces an equational

* Fax: 0824-24-0756.

E-mail address: arai@mis.hiroshima-u.ac.jp (T. Arai)

PII: S0168-0072(99)00041-X

system *ALV* based on N_0, and shows that any provable equation in *ALV* can be transformed into boolean sentences so that $\mathscr{F}$ has polysize proofs of these sentences. Nevertheless, it is left open whether *ALV* catches the full power of $\mathscr{F}$.

In [7] Buss shows that an intensionally correct truth definition for boolean sentences can be written as polysize boolean formulae. Thus Frege system also has polysize proofs of partial consistencies of itself. The truth definition utilizes *countings*, *vector summations* and a form of *inductive definitions*. Prior to this Buss [5] shows that Frege system $\mathscr{F}$ has polysize proofs of the propositional pigeonhole principles by showing that $\mathscr{F}$ has an intensionally correct definition of counting. The definition utilizes carry–save–additions. Therefore some nontrivial parts of mathematics are carried in the propositional proof system $\mathscr{F}$. Such a fact has been already found by Cook [15] for the extended Frege system $e\mathscr{F}$. A natural question to be asked is: Is this all for $\mathscr{F}$? Namely are all parts of mathematics included in $\mathscr{F}$ derived from countings?

The former ones, viz. countings and vector summations suffice to develop some metamathematics–arithmetizations in $\mathscr{F}$. It seems to us that the latter ingredient, viz. inductive definition is essential. In fact, the former are derived from the latter. This is shown in Section 2 and is expected: The latter inductive definition corresponds to the evaluation for the computation tree of a predicate in the complexity class *ALOGTIME*. Prior to [7] Buss [6] showed that BSVP (boolean sentence value problem) is in *ALOGTIME* and hence is *ALOGTIME*-complete in a weak reducibility. In view of the result in [6] the task in [7] was to show that the algorithm for $BSVP \in ALOGTIME$ is intensionally correct for $\mathscr{F}$.

In this paper *ALOGTIME* is used as a synonym of (uniform) NC^1. By definition a function f of polynomial growth rate is *ALOGTIME*-computable, denoted $f \in \mathscr{F}$ *ALOGTIME*, iff its *bitgraph* $\{(i,\bar{x})\colon Bit(i, f(\bar{x})) = 1\}$ is in *ALOGTIME*.

When one reads these proofs in $\mathscr{F}$ in [5, 7], it is natural to consider that these are images of mathematical–logical–arithmetical proofs in a system of bounded arithmetic. There may be various ways to formulate preimages, i.e., a system of bounded arithmetic in which proofs in [5, 7] are carried out. In this paper we introduce a system *AID* (alogtime inductive definitions) of bounded arithmetic. The main feature of *AID* is to allow a form of inductive definitions. We show the following results.

1. (cf. Theorem 2.1). Bounded vector summation and hence Counting are Σ^b_0-bitdefinable in *AID* (carry–save–addition reduces to inductive definitions in *AID*): If the bitgraph of an $f(i,\bar{y})$ is Σ^b_0-definable in *AID*, then so is the function
$$g(x,\bar{y}) = \sum_{i<|x|} f(i,\bar{y}).$$
2. (cf. Section 4). In *AID* a truth definition for boolean sentences is definable by a Σ^b_0-formula $TRUE(x)$. The definition is nothing but the arithmetical equivalent to Buss' definition in [7] and hence is intensionally correct.
Therefore:
3. (cf. Theorem 4.1). $AID \vdash RFN(\mathscr{F})$, where $RFN(\mathscr{F})$ denotes the reflection schema for $\mathscr{F}$.

4. (cf. Section 6). For each Σ^b_0-formula $B(x)$ there exists a Σ^b_0-bitdefinable function $\varphi_B : x \mapsto \langle B(x) \rangle$ such that $\langle B(x) \rangle$ is a boolean sentence and $AID \vdash B(x) \leftrightarrow TRUE(\langle B(x) \rangle)$.
5. (cf. Theorem 6.1). If $AID \vdash B(x)$ for a Σ^b_0-formula $B(x)$, then $AID + \Sigma^b_0\text{-CA} \vdash \mathscr{F} \vdash^{p|x|} \langle B(x) \rangle$ for a polynomial $p|x|$. Thus

$$AID \vdash B(x) \Leftrightarrow AID + \Sigma^b_0\text{-CA} \vdash \mathscr{F} \vdash^{p|x|} \langle B(x) \rangle \text{ for a polynomial } p|x|.$$

6. (cf. Theorem 8.2). A predicate is in *ALOGTIME* iff it is Σ^b_0-definable in *AID* iff it is Δ^b_1-definable in *AID*.

The aim of *AID* is to capture, calibrate and draw the line to mathematical–arithmetical power of Frege system $\mathscr{F}$. By establishing what can be done in $\mathscr{F}$ mathematically, we hope to find a right candidate of hard tautologies for $\mathscr{F}$ cf. [3, 17], to specify what kind of nonstandard models is to be considered in order to prove superpolynomial lowerbounds for hard tautologies for $\mathscr{F}$, cf. $I\Delta_0(f)$ vs. constant depth Frege systems and nonstandard model of *PA* in Ajtai [1], and to find a bounded arithmetic for constant depth threshold gates TC^0 in order to compare *AID*.

Let us explain contents of sections.

In Section 1 the bounded arithmetic *AID* is defined. First we introduce a base language L_{BA} of weak bounded arithmetics and then the language of *AID* is obtained by adding predicate constants for inductively defined predicates. Also we recall some axiom schemata.

In Section 2 we show first that inductive definitions along quadtree (4-branching tree), iterated inductive definitions and simultaneous inductive definitions reduce to a single inductive definition available in *AID*. Using these we show next that vector summations are Σ^b_0-bitdefinable in *AID*.

In Section 3 we show that each predicate in *ALOGTIME* is Σ^b_0-definable in *AID*.

In Section 4 we examine Buss' propositional consistency proof of Frege systems in [7] and verify that it is formalizable in *AID*.

In Section 5 *stratifications* of formulae are defined. These are in essence to interpret first-order formulae as second-order formulae. In later sections we need these.

In Section 6 we show that any Σ^b_0-theorem in *AID* yields true boolean sentences of which $\mathscr{F}$ has polysize proofs.

In Section 7 we introduce some systems of bounded arithmetic in the language L_{BA}, i.e., without inductively defined predicates which are equivalent to *AID*.

In Section 8 we show that Σ^b_1-consequences in, e.g., $AID + \Sigma^b_0\text{-CA}$ can be realized by a Σ^b_0-set $\{i < p|\bar{x}|: A(\bar{x}, i)\}$.

In Section 9 we show that $AID + \Sigma^b_0\text{-CA}$ is equivalent to Clote's *ALV* in the sense that there exist Σ^b_1-faithful interpretations between $AID + \Sigma^b_0\text{-CA}$ and a quantified theory *QALV*. Hence we see that Clote's *ALV* proves the soundness of Frege system, Corollary 9.1.

There are now several theories besides *ALV* which are designed for Frege systems in [12] and for *ALOGTIME* in [13, 14, 19]. It is open to us whether these are equivalent to *AID*.

The results in Section 1–8 of this paper were contained in a handwritten note ‘Frege System, *ALOGTIME* and Bounded Arithmetic’ written in Dec. 1991. Section 9 is augmented here.

1. The bounded arithmetic *AID*

In this section we introduce the bounded arithmetic *AID*.

1.1. A base language

Function constants in the *language* L_{BA} of weak bounded arithmetic are 0 (zero), 1 (one), $+$ (addition), $\lfloor x/2 \rfloor$ (half), $|x|$ (length), $x\# y$ (smash), $x \cdot 2^{|y|}$ (padding), $x \dot{-} y$ (modified subtraction) and $x[i,j)$ (part). Let x_k denote the kth digit of x in binary representation. Then the part function $x[i,j)$ is defined to be the string $x_{j-1}\cdots x_i$ from ith digit x_i to $(j-1)$th digit x_{j-1}:

$$x[i,j) = \sum_{i\leqslant k<j} x_k \cdot 2^{k-i} \quad \text{for } x = \sum_k x_k \cdot 2^k.$$

Clearly

$$|x[i,j)| = \min\{j,|x|\} \dot{-} i.$$

Thus L_{BA} is obtained from the language of S_2^i in [4] by deleting the multiplication $x \cdot y$ and adding three functions $x \cdot 2^{|y|}$, $x \dot{-} y, x[i,j)$. Further the successor function Sx is replaced by $x+1$.

From these, familiar functions are defined as follows:

1. $|x|_0 = x$ and $|x|_{n+1} = ||x|_n|$. Also $\|x\| = |x|_2$.
2. $Bit(i,x) = x[i,i+1)$ (ith digit)
3. $MSP(x,i) = x[i,|x|)$ (most significant part)
4. $x \subseteq_h y \Leftrightarrow_{df} MSP(y,|y| \dot{-} |x|) = x$ (x is a head segment of y).
5. $|x| \cdot |y| = |x\# y| \dot{-} 1$.

   Multiplication for small numbers is definable as follows:

$$multi(i,j,x,y) = \begin{cases} i\cdot j = |x[0,i)| \cdot |y[0,j)| & \text{if } i \leqslant |x| \ \&\ j \leqslant |y|, \\ 0 & \text{otherwise.} \end{cases}$$

6. *Successor functions xi in binary notation*: Put

$$xi =_{df} s_i x = 2\cdot x + i = x + x + i \quad \text{for } i<2.$$

Remark. Let us explain why we delete the multiplication $x \cdot y$ from the list of given functions in weak bounded arithmetic. First of all the bitgraph G_{M} of $x \cdot y$ is Σ_0^{b}-definable in the language of the bounded arithmetic *AID* defined below, cf. the end of Section 2. This is an expected result since G_{M} is in *ALOGTIME* and *AID* is designed so that it captures the complexity class. Although an equivalent theory would be obtained by adding the function $x \cdot y$ as primitive one, we do not want to do so. The reasons are twofold: an aesthetic one and a technical one.

First the bitgraph G_{M} of multiplication is a hard predicate to define. In fact, we see that G_{M} is TC^0-complete under a weak reducibility from Buss [8]. In axiomatizing a bounded arithmetic capturing a complexity class, it would be better to begin with a list of 'simple' functions and predicates and to single out a few defining principles by which any predicate in the class can be defined from the list. At present it is still open whether $TC^0 =? ALOGTIME$, and hence G_{M} may be a complete predicate for *ALOGTIME*. It would be better not to include such a function or predicate in the language of the theory in which the complexity class is analysed.

Second in translating an arithmetical formula in a resticted form into boolean formulae, each variable x is regarded to code a sequence $\bar{p} = \{p_i;\ i<|x|\}$ of propositional atoms, cf. Section 6. In doing so, we rewrite the arithmetical formula in an equivalent one in normal form (called a *stratified formula*, cf. Section 5.) in which the variable x occurs essentially only in the form $Bit(i,x) = 1$. Then the latter is translated in an atom p_i. We first need to define the bitgraph of each function symbol as a stratified formula. Therefore, if the hard function $x \cdot y$ to be computed is in the list of function symbols, then the translation would be complicated to define: structures of boolean formulae would be remotely related to logical form of the arithmetical formula.

We encode a word $i_{k-1} \cdots i_0 \in \{0,1\}^*$ by *attaching the leading marker* 1:

$$\lceil i_{k-1} \cdots i_0 \rceil = 1 i_{k-1} \cdots i_0 \quad \text{(in binary notation).} \tag{1}$$

For example $\lceil \varepsilon \rceil = 1$ for the empty word ε. Using this encoding *concatenation* on words is defined as follows:

$$x * y = x \cdot 2^{|y| \dot{-} 1} + y[0, |y| \dot{-} 1).$$

If $x = \lceil i_{k-1} \cdots i_0 \rceil$, $y = \lceil j_{l-1} \cdots j_0 \rceil$, then $x * y = \lceil i_{k-1} \cdots i_0 j_{l-1} \cdots j_0 \rceil$. Clearly, $|x * y| = |x| + |y| \dot{-} 1$ for $y > 0$.

BASIC denotes the set of basic axioms for constants in L_{BA}. These are obtained from basic axioms in [4, p. 31], by deleting axioms mentioning multiplication and adding the following axioms:

1. $j \leqslant i \rightarrow x[i,j) = 0$; $i < j \rightarrow x[i+1, j+1) = \lfloor \frac{x}{2} \rfloor [i,j)$,
2. $(2x)[0,j+1) = 2 \cdot (x[0,j)) \& (2x+1)[0,j+1) = 2 \cdot (x[0,j)) + 1$,
3. $x \cdot 2^{|0|} = x$; $y \neq 0 \rightarrow x \cdot 2^{|y|} = x \cdot 2^{|\lfloor \frac{y}{2} \rfloor|} + x \cdot 2^{|\lfloor \frac{y}{2} \rfloor|}$,
4. $x \leqslant y \rightarrow x \dot{-} y = 0$; $x \geqslant y \rightarrow (x+1) \dot{-} y = (x \dot{-} y) + 1$.

When the language is expanded to include a set $\mathscr{X}$ of predicate constants, then *BASIC* is assumed to include the equality axiom for constants in $\mathscr{X}$. The expanded language is denoted $L_{\mathrm{BA}}(\mathscr{X})$.

Quantifiers of the form $Qx \leqslant t$ ($Q \in \{\forall, \exists\}$) are said to be *bounded quantifiers* while quantifiers of the form $Qx \leqslant |t|$ ($Q \in \{\forall, \exists\}$) are *sharply bounded quantifiers*.

Classes of Σ_i^{b} formulae and Π_i^{b} formulae are defined as in [4]. Also a formula A is in $s\Sigma_i^{\mathrm{b}}$ (*strict* Σ_i^{b}) iff A is in a prenex normal form such that its leading quantifier is a bounded existential quantifier followed by a string of alternating bounded quantifiers with a sharply bounded matrix $B \in \Sigma_0^b$:

$$A \equiv \exists x_1 \leqslant t_1 \quad \forall x_2 \leqslant t_2 \cdots Q x_i \leqslant t_i B$$

$s\Pi_i^b$ is defined dually. Relativized classes $\Sigma_i^b(\mathscr{X})$, $\Pi_i^b(\mathscr{X})$, $s\Sigma_i^b(\mathscr{X})$ and $s\Pi_i^b(\mathscr{X})$ are defined analogously. $\Sigma_i^b(\mathscr{X})$, etc. are denoted by $\Sigma_i^b(L)$ for $L = L_{\mathrm{BA}}(\mathscr{X})$.

Σ_0^b-LIND denotes the following axiom schema:

$$A(0) \wedge \forall y < |x| (A(y) \rightarrow A(y+1)) \rightarrow A(|x|)$$

for $A \in \Sigma_0^b$.

A base fragment Σ_0^b-LIND of bounded arithmetic: its language is L_{BA} and its axioms are *BASIC*, the axiom schema Σ_0^b-LIND and the *Bit Extensionality Axiom*.

$$|x| = |y| \ \& \ \forall i < |x| \ (Bit(i,x) = Bit(i,y)) \rightarrow x = y.$$

A term of the form $\ell\bar{x} = \sum_{i=1}^n c_i x_i + d$ for some constants, i.e, numerals c_i $(1 \leqslant i \leqslant n)$, d is said to be a *linear form* (in a sequence $\bar{x} = x_1, \ldots, x_n$ of variables). Also $\ell\|\bar{x}\|$ denotes $\sum_{i=1}^n c_i \|x_i\| + d$.

Let t be either a polynomial $p|\bar{x}| = p(|x_1|, \ldots, |x_n|)$ or a linear form $\ell\|\bar{x}\|$. Assume that a variable y does not occur in t. Then $\forall |y| \leqslant t B(y)$ denotes the formula $\forall y \leqslant 2^t (|y| \leqslant t \rightarrow B(y))$. If t is a polynomial $p|\bar{x}|$, then the quantifier $\forall |y| \leqslant t$ is a bounded quantifier since the smash function # is in L_{BA}. If t is a linear form, then it is a sharply bounded quantifier. The existential one $\exists |y| \leqslant t$ is defined dually.

1.2. Some axiom schemata

Put

$$i \in x \quad \Leftrightarrow_{\mathrm{df}} \quad Bit(i,x) = 1.$$

Using this we define analogues of some axiom schemata in second-order arithmetic. Let $L_{\mathrm{BA}}(\mathscr{X})$ be an expanded language for a set $\mathscr{X}$ of predicate constants and Φ a set of formulae in $L_{\mathrm{BA}}(\mathscr{X})$.

Φ-CA denotes the axiom schema:

$$\exists |y| \leqslant p|\bar{x}| \forall i < p|\bar{x}| \ (i \in y \leftrightarrow B(i;\bar{x}))$$

for each polynomial $p|\bar{x}|$ and each formula $B(i;\bar{x}) \in \Phi$.

Φ-LCA denotes the axiom schema:

$$\exists |y| \leqslant \ell\|\bar{x}\| \forall i < \ell\|\bar{x}\| \ (i \in y \leftrightarrow B(i;\bar{x}))$$

for each linear form $\ell\|\bar{x}\|$ and each formula $B(i;\bar{x}) \in \Phi$.

$\Delta_1^b(L_{\mathrm{BA}}(\mathscr{X}))$-CA denotes the axiom schema

$$\forall i (B(i;\bar{x}) \leftrightarrow \neg C(i;\bar{x})) \rightarrow \exists |y| \leqslant p|\bar{x}| \quad \forall i < p|\bar{x}| \ (i \in y \leftrightarrow B(i;\bar{x}))$$

for each polynomial $p|\bar{x}|$ and each $B(i;\bar{x}), C(i;\bar{x}) \in \Sigma_1^b(L_{\mathrm{BA}}(\mathscr{X}))$.

Φ-AC (or Φ-replacement) denotes the axiom schema.
$\forall i < p|\bar{x}| \exists |y| \leqslant q|\bar{x}|\, B(i, y; \bar{x}) \rightarrow \exists |z| \leqslant p|\bar{x}| \cdot q|\bar{x}|\ \forall i < p|\bar{x}|\, B(i, z_i; \bar{x})$
for polynomials $p|\bar{x}|, q|\bar{x}|$ and each $B \in \Phi$, where $z_i = z[q|\bar{x}| \cdot i, q|\bar{x}| \cdot (i+1))$.
$\Delta^{\mathrm{b}}_1(L_{\mathrm{BA}}(\mathscr{X}))$-LIND denotes the axiom schema:

$$\forall y\ (A(y) \leftrightarrow \neg B(y)) \rightarrow A(0) \wedge \forall y < |x|(A(y) \rightarrow A(y+1)) \rightarrow A(|x|)$$

for $A, B \in \Sigma^{\mathrm{b}}_1(L_{\mathrm{BA}}(\mathscr{X}))$.
The following lemma is a folklore, e.g., cf. [18].

Lemma 1.1. *Over* $\Sigma^{\mathrm{b}}_0(L_{\mathrm{BA}}(\mathscr{X}))$-LIND *we have*
1. $\Sigma^{\mathrm{b}}_0(L_{\mathrm{BA}}(\mathscr{X}))$-LCA.
2. $\Sigma^{\mathrm{b}}_0(L_{\mathrm{BA}}(\mathscr{X}))$-AC *proves* $\Sigma^{\mathrm{b}}_0(L_{\mathrm{BA}}(\mathscr{X}))$-CA, $\Sigma^{\mathrm{b}}_1(L_{\mathrm{BA}}(\mathscr{X}))$-AC, $\Delta^{\mathrm{b}}_1(L_{\mathrm{BA}}(\mathscr{X}))$-CA *and* $\Delta^{\mathrm{b}}_1(L_{\mathrm{BA}}(\mathscr{X}))$-LIND.

Definition 1.1 (*Bitdefinable functions*)**.** Let $f(\bar{x})$ be a function of polynomial growth rate, and L_T the language of a theory T. We say that $f(\bar{x})$ is Σ^{b}_0-bitdefinable in T if its bitgraph $\{(i,\bar{x})\colon Bit(i, f(\bar{x})) = 1\}$ is definable by a $\Sigma^{\mathrm{b}}_0(L_T)$-formula, i.e., there exists a $\Sigma^{\mathrm{b}}_0(L_T)$-formula $A_f(i,\bar{x})$ such that in the standard model $A_f(i,\bar{x}) \leftrightarrow Bit(i, f(\bar{x})) = 1$.
We say that $f(\bar{x})$ is Σ^{b}_0*-definable in the theory* T if its graph is definable by a $\Sigma^{\mathrm{b}}_0(L_T)$-formula $G_f(y,\bar{x})$ in the standard model, and $T \vdash \forall \bar{x}\, \exists! y\, G_f(y,\bar{x})$.

Lemma 1.2. 1. *Suppose a function* $f(\bar{x})$ *is* Σ^{b}_0*-bitdefinable in a theory* T. *If the axiom schema* $\Sigma^{\mathrm{b}}_0(L_T)$-CA *and the Bit Extensionality axiom are provable in* T, *then* $T \vdash \forall \bar{x} \exists! y(y = \{i < p|\bar{x}|\colon A_f(i,\bar{x})\})$ *for a polynomial* p *with* $|f(\bar{x})| \leqslant p|\bar{x}|$. *Therefore,* $f(\bar{x})$ *is* Σ^{b}_0*-definable in the theory* T.
2. *Let* $f(\bar{x})$ *be a function of a small value, i.e.,* $|f(\bar{x})| \leqslant \ell \|\bar{x}\|$ *for a linear form* ℓ. *Then for any theory* T *containing the base fragment* $\Sigma^{\mathrm{b}}_0(L_T)$-LIND, *the function* $f(\bar{x})$ *is* Σ^{b}_0*-definable in* T *iff it is* Σ^{b}_0*-bitdefinable in* T. *In this case we can use freely the function symbol* f *in* Σ^{b}_0*-formula.*

Proof. We show Lemma 1.2.2. If $f(\bar{x})$ is Σ^{b}_0-definable in T, then $A_f(i,\bar{x}) \leftrightarrow \exists |y| \leqslant \ell \|\bar{x}\| (G_f(y,\bar{x}) \& Bit(i, y) = 1)$ for a $\Sigma^{\mathrm{b}}_0(L_T)$-graph G_f of f. Hence f is Σ^{b}_0-bitdefinable in T. The converse direction follows from Lemmas 1.2.1 and 1.1.1. $\square$

1.3. The bounded arithmetic AID

Now we define a bounded arithmetic *AID*.

Definition 1.2. The *language* L_{AID} of *AID*: Given a linear form $\ell \|\bar{x}\| \mathrm{in} \|\bar{x}\| = \|x_1\|, \ldots, \|x_n\|$, Σ^{b}_0-formulae $B(\bar{x}, p), \bar{D}(\bar{x}, p) = D_1(\bar{x}, p), \ldots, D_m(\bar{x}, p)$ in L_{BA} and a boolean propositional formula $I(\bar{d}, p_0, p_1)$ in atoms $\bar{d} = d_1, \ldots, d_m$ and p_0, p_1 we introduce an $(n+1)$-*ary predicate constant* $A^{\ell, B, \bar{D}, I}$. Then L_{AID} is defined to be the expanded language of

L_{BA} having the predicate constant $A^{\ell,B,\bar{D},I}$ for each such items $\ell,B,\bar{D},I$. When no confusion likely occurs, we write A for $A^{\ell,B,\bar{D},I}$.

AID is obtained from $\Sigma^{\mathrm{b}}_0(L_{AID})$-LIND, i.e., Σ^{b}_0-LIND in the language L_{AID} by adding the following *axioms for the newly introduced predicate* $A=A^{\ell,B,\bar{D},I}$:
(A.0) $A(\bar{x},p)\to 0\neq|p|\leqslant\ell\|\bar{x}\|$,
(A.1) $0\neq|p|=\ell\|\bar{x}\|\to[A(\bar{x},p)\leftrightarrow B(\bar{x},p)]$,
(A.2) $0\neq|p|<\ell\|\bar{x}\|\to[A(\bar{x},p)\leftrightarrow I(\bar{D}(\bar{x},p),A(\bar{x},p0),A(\bar{x},p1))]$,
where the RHS $I(\bar{D}(\bar{x},p),A(\bar{x},p0),A(\bar{x},p1))$ denotes the result $I(D_1(\bar{x},p),\dots,D_m(\bar{x},p),A(\bar{x},p0),A(\bar{x},p1))$ of simultaneous replacement of the atoms $\bar{d}=d_1,\dots,d_m$, p_0,p_1 by the formulae $D_1(\bar{x},p),\dots,D_m(\bar{x},p),A(\bar{x},p0)$ and $A(\bar{x},p1)$ in the boolean formula I. Intuitively p ranges over binary strings of length $<\ell\|\bar{x}\|$, cf. the encoding (1).

(A.0)–(A.2) give the *inductive definition of the predicate* $A=A^{\ell,B,\bar{D},I}$. To decide $A(\bar{x},p)$ for $0\neq|p|\leqslant\ell\|\bar{x}\|$ build a binary tree of depth $\ell\|\bar{x}\|-|p|$. The sons of the node $A(\bar{x},p)$ are $A(\bar{x},p0),A(\bar{x},p1)$ and we put a 'gate' $I(\bar{D}(\bar{x},p),p_0,p_1)$ there. p in $A(\bar{x},p)$ is a *clock*, i.e., it tells us the time when we stop to calculate the truth values, namely $|p|=\ell\|\bar{x}\|$ by (A.1).

The point is that we can decide $A(\bar{x},p)$ from the sons $A(\bar{x},p0),A(\bar{x},p1)$ propositionally. The essence of the clause (A.2) is that previous stages are not quantified at all in the RHS.

We can assign truth values $A^{\ell,B,\bar{D},I}(\bar{x},p)$ for $p=0$ and for $|p|>\ell\|\bar{x}\|$ in an arbitrary manner since we need only $\{A(\bar{x},p)\colon 0\neq|p|\leqslant|\ell\|\bar{x}\|\}$.

It is straightforward to see the following proposition, cf. Theorem 4 in [6].

Proposition 1.1. *Each Σ^{b}_0-formula in L_{AID} defines a predicate in ALOGTIME.*

2. Inductive definitions and vector summations in *AID*

In this section we show first that inductive definitions along quadtree (4-branching tree), iterated inductive definitions and simultaneous inductive definitions reduce to a single inductive definition available in *AID*. Using these we show next that vector summations are Σ^{b}_0-bitdefinable in *AID*.

Lemma 2.1 (Tree induction). *For a linear form $\ell\|\bar{x}\|$ and a Σ^{b}_0-formula B in L_{AID}, we have in AID*

$$\forall|p|\leqslant\ell\|\bar{x}\|\left[(0\neq|p|=\ell\|\bar{x}\|\to B(p))\,\&\left(0\neq|p|<\ell\|\bar{x}\|\,\&\bigwedge_{i<2}B(pi)\to B(p)\right)\right]$$
$$\to\forall|p|\leqslant\ell\|\bar{x}\|(0\neq|p|\to B(p)).$$

Proof. Apply Σ^{b}_0-LIND to the following Σ^{b}_0 $C(u)$:

$$C(u)\ \Leftrightarrow_{\mathrm{df}}\ \forall|p|\leqslant\ell\|\bar{x}\|(0\neq|p|\,\&\,|p|+u\geqslant\ell\|\bar{x}\|\to B(p)).\qquad\square$$

In the following lemma let $A = A^{\ell,B,\bar{D},I}$ be a predicate defined by (A.0)–(A.2). Let C^+, C^- be $\Sigma^{\mathrm{b}}_0(L_{AID})$ formulae. By separating positive, negative occurrences of atoms p_0, p_1 in the boolean formula I, we set

$$I^+(\bar{d}, p_0^+, p_0^-, p_1^+, p_1^-) \leftrightarrow I(\bar{d}, p_0, p_1).$$

The superscript $+$ $[-]$ designates the positive [negative] occurrences. Put

$$I^-(\bar{d}, p_0^+, p_0^-, p_1^+, p_1^-) \Leftrightarrow_{\mathrm{df}} \neg I^+(\bar{d}, p_0^-, p_0^+, p_1^-, p_1^+).$$

Let IH denote the formula

$$\begin{aligned}
&\forall |p| \leqslant \ell \|\bar{x}\| [\\
&\{0 \neq |p| = \ell\|\bar{x}\| \to (B(\bar{x}, p) \to C^+(p)) \,\&\, (\neg B(\bar{x}, p) \to C^-(p))\} \,\&\\
&\{0 \neq |p| < \ell\|\bar{x}\| \to\\
&(I^+(\bar{D}(\bar{x}, p), C^+(p0), C^-(p0), C^+(p1), C^-(p1)) \to C^+(p)) \,\&\\
&(I^-(\bar{D}(\bar{x}, p), C^+(p0), C^-(p0), C^+(p1), C^-(p1)) \to C^-(p))\}].
\end{aligned}$$

Then we see the following lemma from Lemma 2.1.

Lemma 2.2 (Proof by tree induction). *AID proves that*

$$IH \to \forall |p| \leqslant \ell\|\bar{x}\| [0 \neq |p| \to (A(\bar{x}, p) \to C^+(p)) \,\&\, (\neg A(\bar{x}, p) \to C^-(p))].$$

Lemma 2.3 (Inductive definitions along quadtrees). *Let $\ell\|\bar{x}\|$ be a linear form, $B, \bar{D}$ Σ^{b}_0-formulae in L_{BA} and $I(\bar{d}, p_{00}, p_{01}, p_{10}, p_{11})$ a boolean formula. Define a predicate A inductively by*

(A.0) *$A(\bar{x}, p) \to 0 \neq |p| \leqslant 2\ell\|\bar{x}\| + 1 \,\&\, |p|$ is odd.*

(A.1) *The case $0 \neq |p| = 2\ell\|\bar{x}\| + 1 : A(\bar{x}, p) \leftrightarrow B(\bar{x}, p)$.*

(A.2) *The case $0 \neq |p| < 2\ell\|\bar{x}\| + 1 \,\&\, |p|$ is odd:*

$$\begin{aligned}
A(\bar{x}, p) &\leftrightarrow I(\bar{D}(\bar{x}, p), \{A(\bar{x}, pij) : \ i, j < 2\})\\
&\leftrightarrow I(\bar{D}(\bar{x}, p), A(\bar{x}, p00), A(\bar{x}, p01), A(\bar{x}, p10), A(\bar{x}, p11)),
\end{aligned}$$

where $pij = 2(2p + i) + j$.

Then A can be Σ^{b}_0-defined in AID.

Proof. For a formula F and $i < 2$ put

$$F^i = \begin{cases} F, & i = 1, \\ \neg F, & i = 0. \end{cases} \tag{2}$$

For $i, j < 2$ put $k(ij) = Bit(2i + j, k)$.

First write the boolean formula I in a DNF (disjunctive normal form)

$$\bigvee \{C_k : \ k < 2^4\} \leftrightarrow I(\bar{D}, \{A(\bar{x}, pij) : \ i, j < 2\}) \tag{3}$$

such that each disjunct C_k is of the form

$$C_k \equiv I_k(p) \wedge \bigwedge \{A(\bar{x}, pij)^{k(ij)}:\ i,j<2\} \tag{4}$$

and the predicate A does not occur in $I_k(p)$.

Now the 4-branching regress (A.2) is simulated by a tree of depth 6: first construct a $\vee$-tree of depth 4 corresponding to $\bigvee$ in the DNF (3) and then construct trees of depth 2 below each leaf of the $\vee$-tree to handle $\bigwedge$ in (4). We define $A(\bar{x}, p)$ using a new clock q which is divided by 6-digits.

Define a predicate A' inductively as follows:
(A$'$.0)

$$A'(\bar{x}, p, q) \to |p| \text{ is odd} \,\&\, |p| + q_0 \leqslant 2\ell + 1 \,\&\, 0 \neq |q| \leqslant \ell',$$

where $\ell = \ell\|\bar{x}\|$, $\ell' = 1 + 6(\ell - \lfloor |p|/2 \rfloor)$ and

$$q_0 = \begin{cases} 2(\lfloor \frac{|q| \dot{-} 1}{6} \rfloor) & \text{if } |q| \dot{-} 1 \equiv 0 \pmod 6, \\ 2(\lfloor \frac{|q| \dot{-} 1}{6} \rfloor) + 2 & \text{otherwise.} \end{cases}$$

In the following assume $|p|$ is odd & $|p| + q_0 \leqslant 2\ell + 1 \,\&\, 0 \neq |q| \leqslant \ell'$, the RHS of (A$'$.0).
(A$'$.1) The case $|q| < \ell' \,\&\, |q| \equiv 1,2,3,4 \pmod 6$:

$$A'(\bar{x}, p, q) \leftrightarrow A'(\bar{x}, p, q0) \vee A'(\bar{x}, p, q1).$$

(A$'$.2) The case $|q| < \ell' \,\&\, |q| \equiv 5 \pmod 6) : A'(\bar{x}, p, q)$ iff for some $k < 2^4$

$$\exists |r| < \ell [R_2(r,q) \,\&\, I_k(p * r)] \,\&\, A'(\bar{x}, p, q0) \,\&\, A'(\bar{x}, p, q1) \,\&\, k = q[0,4),$$

where $R_2(r,q)$ denotes the formula

$$|q| = 3(|r| \dot{-} 1) + 5 \,\&\, \forall j < 2 \forall i < \left\lfloor \frac{|q|}{6} \right\rfloor (Bit(2i + j, r) = Bit(6i + j + 4, q)).$$

By Lemma 1.1.1, Σ^{b}_0-LCA and Bit Extensionality axiom such a number r is uniquely determined from q.
(A$'$.3) The case $|qi| < \ell' \,\&\, |q| \equiv 5 \pmod 6$ for an $i < 2$:

$$A'(\bar{x}, p, qi) \leftrightarrow \bigwedge \{A'(\bar{x}, p, qij)^{q(ij)}:\ j < 2\},$$

where $F^{q(ij)} \Leftrightarrow_{\mathrm{df}} (q(ij) = 1 \,\&\, F) \vee (q(ij) = 0 \,\&\, \neg F)$ and $q(ij) = Bit(2i + j, q)$.
(A$'$.4) The case $|1q| = \ell' = 1 + 6(\ell - \lfloor |p|/2 \rfloor)$:

$$A'(\bar{x}, p, 1q) \leftrightarrow \exists |r| \leqslant \ell [R_4(r,q) \,\&\, B(\bar{x}, p * r)],$$

where $R_4(r,q)$ denotes the formula

$$|q| = 3(|r| \dot{-} 1) \,\&\, \forall j < 2 \forall i < \left\lfloor \frac{|q|}{6} \right\rfloor (Bit(2i + j, r) = Bit(6i + j, q)).$$

By induction on $6(\ell - \lfloor |p|/2 \rfloor) - |q|$ we have the following Claim 2.1:

Claim 2.1.

$$AID \vdash |q| \equiv 0 \pmod 6 \,\&\, R_4(r,q) \to [A'(\bar{x}, p * r, 1) \leftrightarrow A'(\bar{x}, p, 1q)],$$

where $R_4(r,q)$ denotes the formula in (A$'$.4)

By Claim 2.1 we have for $i,j < 2$ and $1k = 2^4 + k$,

$$\forall k < 2^4 (A'(\bar{x}, pij, 1) \leftrightarrow A'(\bar{x}, p, 1kij)). \tag{5}$$

Thus by putting

$$A(\bar{x}, p) \;\Leftrightarrow_{\mathrm{df}}\; A'(\bar{x}, p, 1),$$

we get the defining axioms (A.0)–(A.2). For example to see (A.2) assume $0 \neq |p| < 2\ell\|\bar{x}\| + 1 \,\&\, |p|$ is odd. Then we have $6 < 1 + 6(\ell - \lfloor |p|/2 \rfloor)$ by $|p|$ is odd and hence $\lfloor |p|/2 \rfloor < \ell$. Using the defining axioms (A$'$.1)–(A$'$.3) of A', (5), (3) and (4) we have

$$\begin{aligned}
A(\bar{x}, p) &\Leftrightarrow_{\mathrm{df}} A'(\bar{x}, p, 1) \\
&\leftrightarrow \bigvee \{ I_k(p) \,\&\, \bigwedge \{ A'(\bar{x}, p, 1kij)^{k(ij)} : \; i,j < 2 \} : k < 2^4 \} \\
&\leftrightarrow \bigvee \{ I_k(p) \,\&\, \bigwedge \{ A'(\bar{x}, pij, 1)^{k(ij)} : \; i,j < 2 \} : k < 2^4 \} \\
&\leftrightarrow I(\bar{D}(\bar{x}, p), \{ A'(\bar{x}, pij, 1) : \; i,j < 2 \}) \\
&\Leftrightarrow I(\bar{D}(\bar{x}, p), \{ A(\bar{x}, pij) : \; i,j < 2 \}). \qquad \square
\end{aligned}$$

By Lemma 2.3 inductive definitions along K-branching trees for each constant $K \geqslant 2$ are Σ^b_0-definable in AID.

Lemma 2.4 (Iterated inductive definitions). *Let B and $\bar{D}$ be Σ^b_0-formulae in L_{AID}. Namely inductively defined predicates may occur in these formulae. Let $A = A^{\ell, B, \bar{D}, I}$ be an inductively defined predicate defined by* (A.0)–(A.2) *in Definition* 1.2 *from a linear form $\ell\|\bar{x}\|$ and a boolean formula I. Then A is Σ^b_0-definable in AID.*

Proof. (*Step* 1): First we push down inductively defined predicates occurring in $\bar{D}$ to the terminal condition B. For simplicity assume that the number m of atoms $d_1, \ldots, d_m$ in I is 2 and let $I = I(d_2, d_3, p_0, p_1)$ and $\bar{D}(\bar{x}, p) = D_2(\bar{x}, p), D_3(\bar{x}, p)$. Further assume $\ell\|\bar{x}\| \neq 0$, i.e., $\ell\|\bar{x}\| = \sum c_i \|x_i\| + d$ with $d \neq 0$.

We say that p *contains a digit* 2 *or* 3 iff $p[i, i+2) \in \{2,3\}$ for some even $i < |p|$. Define a predicate A' by induction along a quadtree as follows:

(A$'$.0) $A'(\bar{x}, p) \to |p|$ is odd $\&\, |p| \leqslant 2\ell\|\bar{x}\| - 1$.

In the following assume that the RHS $|p|$ is odd $\&\, |p| \leqslant 2\ell\|\bar{x}\| - 1$ of (A$'$.0).

(A$'$.1) The case $|p| < 2\ell\|\bar{x}\| - 1$ and p does not contain digit 2 or 3:

$$A'(\bar{x}, p) \leftrightarrow I(A'(\bar{x}, p10), A'(\bar{x}, p11), A'(\bar{x}, p00), A'(\bar{x}, p01)).$$

(A$'$.2) The case $|p|<2\ell\|\bar{x}\|-1$ and p contains a digit 2 or 3:

$$A'(\bar{x},p)\leftrightarrow A'(\bar{x},p00).$$

(A$'$.3) The case $|p|=2\ell\|\bar{x}\|-1$ and p does not contain digit 2 or 3:

$$\begin{aligned}&A'(\bar{x},p)\\&\quad\leftrightarrow\exists|q|\leqslant\ell\|\bar{x}\|[|q|=\ell\|\bar{x}\|-1\,\&\,\forall i<|q|\dot{-}1(Bit(i,q)=Bit(2i,p))\,\&\,B(\bar{x},q)].\end{aligned}$$

(A$'$.4) The case $|p|=2\ell\|\bar{x}\|-1$ and p contains a digit 2 or 3:
(4.1) $A'(\bar{x},p)\to\exists!i<|p|\,[i$ is even $\&\ p[i,i+2)\in\{2,3\}]$.
Let $i<|p|$ denote the unique i such that i is even $\&\ p[i,i+2)\in\{2,3\}$.
(4.k) The case $p[i,i+2)=k\ (k\in\{2,3\})$:

$$\begin{aligned}A'(\bar{x},p)\leftrightarrow\exists|q|\leqslant\ell\|\bar{x}\|[&|q|=\ell\|\bar{x}\|-1-\left\lfloor\frac{i}{2}\right\rfloor\\&\&\forall j<|q|\dot{-}1(Bit(j,q)=Bit(2j+i+2,p))\,\&\,D_k(\bar{x},q)].\end{aligned}$$

Then A' is defined from a boolean formula I', Σ^b_0-formulae $\bar{D}'$ in L_{BA} and a Σ^b_0-terminal condition B' in L_{AID} along a quadtree. From the proof of Lemma 2.3 we see that A' can be defined from a boolean formula I'', Σ^b_0 $\bar{D}''$ in L_{BA} and a Σ^b_0 B'' in L_{AID} along a binary tree.

Further we see easily that for $k<2$,

$$A(\bar{x},q)\leftrightarrow A'(\bar{x},p)\quad\text{and}\quad D_{1k}(\bar{x},q)\leftrightarrow A'(\bar{x},p1k),$$

where $10=2, 11=3$ and p denotes the number such that $|p|=2|q|-1$ and $\forall i<|q|\dot{-}1$ $(Bit(2i,p)=Bit(i,q)\,\&\,Bit(2i+1,p)=0)$.

(*Step* 2): Now we assume that no inductively defined predicate occur in $\bar{D}$. Without loss of generality, we may assume that the Σ^b_0-terminal condition $B(\bar{x},p)$ is in a prenex normal form $\forall|y|\leqslant\ell'\|\bar{x}\|B_0(\bar{x},p,y)$.

Define A' as follows:
(A$'$.1) The case $0\neq|p|=\ell\|\bar{x}\|+\ell'\|\bar{x}\|$:

$$A'(\bar{x},p)\leftrightarrow B_0(\bar{x},p_0,p_1),$$

where $p_0=p[\ell'\|\bar{x}\|,|p|)$ and $p_1=p[0,\ell'\|\bar{x}\|)$.
(A$'$.2) The case $\ell\|\bar{x}\|\leqslant|p|<\ell\|\bar{x}\|+\ell'\|\bar{x}\|$:

$$A'(\bar{x},p)\leftrightarrow A'(\bar{x},p0)\wedge A'(\bar{x},p1).$$

(A$'$.3) The case $0\neq|p|<\ell\|\bar{x}\|$:

$$A'(\bar{x},p)\leftrightarrow I(\bar{D}(\bar{x},p),A'(\bar{x},p0),A'(\bar{x},p1)).$$

Then $|p|=\ell\|\bar{x}\|\to(A'(\bar{x},p)\leftrightarrow\forall|y|\leqslant\ell'\|\bar{x}\|B_0(\bar{x},p,y))$ and hence

$$0\neq|p|\leqslant\ell\|\bar{x}\|\to(A(\bar{x},p)\leftrightarrow A'(\bar{x},p)).$$

The number of sharply bounded quantifiers in the terminal condition B_0 for A' is fewer than one in B for A. Thus, we may assume that the terminal condition $B(\bar{x}, p)$ is an open formula in DNF. By considering a slightly higher tree, cf. proof of Lemma 2.3, we may assume further that the terminal condition is a literal.

(*Step 3*): Now suppose that the terminal condition $B(\bar{x}, p)$ is an atomic formula of the form $A'(\bar{t}(\bar{x}, p), s(\bar{x}, p))$ (or its negation) for some terms $\bar{t}, s$ and an inductively defined predicate A'. Namely A is defined by

(A.2) $0 \neq |p| < \ell\|\bar{x}\| \to [A(\bar{x}, p) \leftrightarrow I(\bar{D}(\bar{x}, p), A(\bar{x}, p0), A(\bar{x}, p1))]$.

(A.1) $0 \neq |p| = \ell\|\bar{x}\| \to [A(\bar{x}, p) \leftrightarrow A'(\bar{t}(\bar{x}, p), s(\bar{x}, p))]$.

While A' is defined from ℓ', I' and some Σ^b_0-formulae $B', \bar{D}'$ in L_{BA} as follows:

(A$'$.1) $0 \neq |p| = \ell'\|\bar{y}\| \to [A'(\bar{y}, p) \leftrightarrow B'(\bar{y}, p)]$.

(A$'$.2) $0 \neq |p| < \ell'\|\bar{y}\| \to [A'(\bar{y}, p) \leftrightarrow I'(\bar{D}'(\bar{y}, p), A'(\bar{y}, p0), A'(\bar{y}, p1))]$.

Define A alternatively as follows:

(A.2) $0 \neq |p| < \ell\|\bar{x}\| \to [A(\bar{x}, p) \leftrightarrow I(\bar{D}(\bar{x}, p), A(\bar{x}, p0), A(\bar{x}, p1))]$.

In the following assume $0 \neq |p| \geqslant \ell\|\bar{x}\|$. Put $p = q * r$ with $|q| = \ell\|\bar{x}\|$ and $r > 0$. Our aim is to have

$$A(\bar{x}, p) \leftrightarrow A'(\bar{t}(\bar{x}, q), s(\bar{x}, q) * r). \tag{6}$$

(a) $A(\bar{x}, p) \to 0 \neq |s(\bar{x}, q)| + |r| \dot{-} 1 \leqslant \ell'\|\bar{t}(\bar{x}, q)\|$.

Assume the RHS $0 \neq |s(\bar{x}, q)| + |r| \dot{-} 1 \leqslant \ell'\|\bar{t}(\bar{x}, q)\|$ of (a).

(b) The case $|s(\bar{x}, q)| + |r| \dot{-} 1 < \ell'\|\bar{t}(\bar{x}, q)\|$:

$$A(\bar{x}, p) \leftrightarrow I'(\bar{D}'(\bar{t}(\bar{x}, q), s(\bar{x}, q) * r), A(\bar{x}, p0), A(\bar{x}, p1)).$$

(c) The case $|s(\bar{x}, q)| + |r| \dot{-} 1 = \ell'\|\bar{t}(\bar{x}, q)\|$:

$$A(\bar{x}, p) \leftrightarrow B'(\bar{t}(\bar{x}, q), s(\bar{x}, q) * r).$$

By (a)–(c) we see (6). Also by $|q| = \ell\|\bar{x}\|$, $|p|$ varies through

$$|p| \leqslant \ell\|\bar{x}\| + \ell'\|\bar{t}(\bar{x}, q)\| - |s(\bar{x}, q)| \leqslant \ell\|\bar{x}\| + \ell''\|\bar{x}\|$$

for some linear form ℓ''. $\square$

Next, we show that simultaneous inductive definitions and vector summations are definable in *AID*.

Lemma 2.5 (Simultaneous induction). *Let $\ell = \ell\|\bar{x}\|$ be a linear form, $B(\bar{x}, \lambda, q)$, $\bar{D}(\bar{x}, \lambda, q)$ Σ^b_0-formulae in L_{AID} and $I(\bar{d}, \{q_{ji}\colon j \leqslant K \,\&\, i < 2\})$ be a boolean formula with a constant K. Define a predicate A inductively by*

(A.0) $A(\bar{x}, \lambda, q) \to 0 \neq |q| \leqslant \ell \,\&\, |\lambda| \leqslant \ell$.

Assume the RHS, $0 \neq |q| \leqslant \ell \,\&\, |\lambda| \leqslant \ell$ in the following.

(A.1) $|q| = \ell \to [A(\bar{x}, \lambda, q) \leftrightarrow B(\bar{x}, \lambda, q)]$.

(A.2) $|q| < \ell \to [A(\bar{x}, \lambda, q) \leftrightarrow I(\bar{D}(\bar{x}, \lambda, q), \{A(\bar{x}, \lambda + j, qi)\colon i < 2 \,\&\, j \leqslant K\})]$.

Then A is Σ^b_0-definable in AID.

Theorem 2.1 (Bounded vector summation). *If $f(i,\bar{y})$ is Σ^b_0-bitdefinable in AID, then so is the function*

$$g(x,\bar{y}) = \sum_{i<|x|} f(i,\bar{y}).$$

Also the defined function g enjoys demonstrably in AID, $g(0,\bar{y})=0$ and for $x \neq 0$

$$g(x,\bar{y}) = g\left(\left\lfloor \frac{x}{2} \right\rfloor, \bar{y}\right) + f(|x| \dot{-} 1, \bar{y}), \tag{7}$$

where $=$ means that these are coextensional.

Corollary 2.1 (Bounded counting). *For each Σ^b_0-formula φ in L_{AID}, the function*

$$C_\varphi(x) = \#\{i < |x| \colon \varphi(i)\}$$

is Σ^b_0-bitdefinable in AID. Also the defined function C_φ enjoys demonstrably in AID, $C_\varphi(0)=0$ and for $x \neq 0$

$$C_\varphi(x) = \begin{cases} C_\varphi\left(\lfloor \frac{x}{2} \rfloor\right) + 1 & \text{if } \varphi(|x|-1), \\ C_\varphi\left(\lfloor \frac{x}{2} \rfloor\right) & \text{otherwise.} \end{cases}$$

Hence by $\Sigma^b_0 - LIND$,

$$\#\{i < |x| \colon \varphi(i)\} = \#\{i < |y| \colon \varphi(i)\} + \#\{i < |x| \dot{-} |y| \colon \varphi(|y|+i)\}. \tag{8}$$

Note that we always have $C_\varphi(x) \leqslant |x|$.

First assuming Lemma 2.5 we show Theorem 2.1 using carry–save–addition to combine four numbers into two, in [5, p. 922].

Proof of Theorem 2.1. For simplicity, we suppress parameters $\bar{y}$. Pick a polynomial H so that $|\sum_{i<|x|} f(i)| \leqslant H(|x|)$. We Σ^b_0-bitdefine the function $g(j,x) = \sum\{f(i) \colon i < \min(j,|x|)\}$.

Define a predicate $A(j,x,\lambda,p)$ by simultaneous recursion in Lemma 2.5 as follows:

(A.0) $A(j,x,\lambda,p) \to 0 \neq |p| \leqslant \|x\| + 1 \,\&\, p < 2^{\|x\|} + |x| \,\&\, \lambda \leqslant 2H(|x|)$.

Assume the RHS of (A.0). In what follows we write

$$\begin{array}{llll} s^p_\lambda \Leftrightarrow_{\mathrm{df}} & s^p_\lambda(j) \Leftrightarrow_{\mathrm{df}} & S(j,x,\lambda,p) \Leftrightarrow_{\mathrm{df}} & A(j,x,2\lambda,p), \\ c^p_\lambda \Leftrightarrow_{\mathrm{df}} & c^p_\lambda(j) \Leftrightarrow_{\mathrm{df}} & C(j,x,\lambda,p) \Leftrightarrow_{\mathrm{df}} & A(j,x,2\lambda+1,p). \end{array}$$

(A.1) The case $|p| = \|x\| + 1$:

$$\begin{array}{l} s^p_\lambda \leftrightarrow A_f(\lambda',p') \wedge p' < j \ (\Leftrightarrow Bit(\lambda', f(p')) = 1 \wedge p' < j), \\ c^p_\lambda \leftrightarrow \bot, \end{array}$$

where $\lambda + \lambda' = H(|x|)$ and $p = 2^{\|x\|} + p'$, $p' < |x|$, i.e., reverse the digits in $f(p')$. Observe that the terminal condition is in $\Sigma^b_0(L_{AID})$.

(A.2) The case $|p| < \|x\| + 1$:

$$s_\lambda^p \leftrightarrow (s_\lambda^{p0} \oplus s_\lambda^{p1} \oplus c_\lambda^{p0}) \oplus (\#\{s_{\lambda+1}^{p0}, s_{\lambda+1}^{p1}, c_{\lambda+1}^{p0}\} \geqslant 2) \oplus c_\lambda^{p1},$$
$$c_\lambda^p \leftrightarrow \#\{s_{\lambda+1}^{p0}, s_{\lambda+1}^{p1}, c_{\lambda+1}^{p0}, \#\{s_{\lambda+2}^{p0}, s_{\lambda+2}^{p1}, c_{\lambda+2}^{p1}\} \geqslant 2, c_{\lambda+1}^{p1}\} \geqslant 2,$$

where $\oplus$ denotes the *excluded or*, and for propositions p, q, r,

$$\#\{p, q, r\} \geqslant 2 \;\Leftrightarrow_{\mathrm{df}}\; (p \wedge q) \vee (q \wedge r) \vee (r \wedge p).$$

For each fixed $p < 2^{\|x\|} + |x|$, $s^p = \{s_\lambda^p\colon \lambda < H(|x|)\}$ gives the reverse binary representation of a number, and similarly for $c^p = \{c_\lambda^p\colon \lambda < H(|x|)\}$.

(A.1) s^p is the reverse binary representation of the number $f(p')$ with $p = 2^{\|x\|} + p'$, $p' < \min(j, |x|)$.

(A.2) In reverse notation, first by carry–save–addition combine three numbers into two, $s^{p0} + s^{p1} + c^{p0} = s' + c'$, and then $s' + c' + c^{p1} = s^p + c^p$ once again by carry–save–addition. Thus $s^{p0} + s^{p1} + c^{p0} + c^{p1} = s^p + c^p$.

Therefore, in reverse notation for s^p, c^p,

$$s^p + c^p = \sum\{f(q')\colon\ p \subseteq_h q = 2^{\|x\|} + q', q' < \min(j, |x|)\}.$$

Let s_λ, c_λ denote the following formulae with $\lambda + \lambda' = H(|x|)$:

$$s_\lambda \leftrightarrow s_{\lambda'}^1\,; c_\lambda \leftrightarrow c_{\lambda'}^1.$$

Let a_λ denote the full addition of s_λ and c_λ:

$$a_\lambda \leftrightarrow s_\lambda \oplus c_\lambda \oplus \exists \eta < \lambda(s_\eta \wedge c_\eta \wedge \forall \rho \in (\eta, \lambda)(s_\rho \oplus c_\rho)).$$

Then a_λ is a Σ_0^{b}-formula such that

$$a_\lambda \leftrightarrow Bit(\lambda, \sum\{f(p')\colon\ p' < \min(j, |x|)\}) = 1$$

for $\lambda \leqslant H(|x|)$.

It remains to show the recursion equation (7) for $g(x) = g(|x|, x)$.

Lemma 2.6. 1. $j < |x| \Rightarrow g(j+1, x) = g(j, x) + f(j)$.

2. *If* $j \leqslant |x| \leqslant |y|$, *then*

$$g(j,x) = \sum\{f(i)\colon\ i < \min(j, |x|)\} = \sum\{f(i)\colon\ i < \min(j, |y|)\} = g(j, y).$$

Proof of Lemma 2.6.

2.6.1. This follows from the fact that in reverse notation for s^p, c^p

$$j < |x| \;\&\; p < 2^{\|x\|} + |x| \Rightarrow s^p(j+1) + c^p(j+1) = s^p(j) + c^p(j) + f^p(j)$$

with

$$f^p(j) = \begin{cases} f(j) & \text{if } p \subseteq_h 2^{\|x\|} + j, \\ 0 & \text{otherwise.} \end{cases}$$

The latter is seen by tree induction on p, Lemma 2.1.

2.6.2. For each p_x with $0 \neq |p_x| \leqslant \|x\|+1$ & $p_x < 2^{\|x\|}+|x|$ let p_y, denote the number such that $|p_y| = |p_x| + (\|y\| \dot{-} \|x\|)$ and $p_y[0, |p_x| \dot{-} 1) = p_x[0, |p_x| \dot{-} 1)$ & $p_y[|p_x| \dot{-} 1, |p_y| \dot{-} 1) = 0$.

Then we see that for $0 \neq |p_x| \leqslant \|x\|+1$ & $p_x < 2^{\|x\|}+|x|$ and for $\lambda \leqslant 2H(|x|)$, $A(j,x,\lambda,p_x) \leftrightarrow A(j,y,\lambda,p_y)$. Furthermore, let $p = p_y = 2^k$ for $p_x = 1$ and $k+1 = \|y\| \dot{-} \|x\| > 0$. We have $A(j,y,\lambda,q) \leftrightarrow \bot$ if q and p are incomparable, i.e., $\neg\exists r \in \{q,p\}$ $(r \subseteq_h q$ & $r \subseteq_h p)$, and hence $A(j,y,\lambda,q) \leftrightarrow A(j,y,\lambda,p)$ for $q \subseteq_h p$. These are seen again by tree induction, Lemma 2.1.

Hence we have shown Lemma 2.6. Now the recursion equation (7) is seen from the lemma: $g(x) = g(j+1,x) = g(j,x) + f(j) = g(j,\lfloor x/2 \rfloor) + f(j) = g(\lfloor x/2 \rfloor) + f(j)$ with $j = |\lfloor x/2 \rfloor|$. This completes the proof of Theorem 2.1 from Lemma 2.5.

Proof of Lemma 2.5. There are $2(1+K)$-sons in the defining tree of $A(\bar{x},\lambda,q)$ where the parameter λ varies. Now the idea is in the shift $(\lambda+j,qi) \Rightarrow (\lambda, qi0^{[K-j]}1^{[j]})$, i.e., storing up variations j in p in unary notation, and discharging this at the terminal node by counting 1's.

We show that $A(\bar{x},\lambda,q)$ is Σ^{b}_0-definable. We define a predicate A' so that, for $|p| \equiv 1 \,(\mathrm{mod}\, 1+K)$,

$$A'(\bar{x},\lambda,p) \leftrightarrow A(\bar{x},\lambda+\#(J),q),$$

where $|p| = 1 + r(1+K)$ and $|q| = 1 + r$, $|J| \leqslant rK$, and their digits are defined by $Bit(i,q) = Bit(K + i(1+K), p)$ for $i < r$, and $Bit(i,J) = Bit(i + \lfloor i/K \rfloor, p)$ for $i < rK$.

$\#(J)$ denotes the number of 1's in the binary representation of the number J:

$$\#(J) = \#\{i < |J|:\ Bit(i,J) = 1\}.$$

Then

$$A(\bar{x},\lambda,1) \leftrightarrow A'(\bar{x},\lambda,1).$$

Such an A' is defined as follows: Each $j \leqslant K$ is coded by $0^{[K-j]}1^{[j]}$ in unary notation. Put $\ell = \ell\|\bar{x}\|$ and $\ell' = 1 + (\ell-1)(1+K)$.

(A$'$.0) $A'(\bar{x},\lambda,p) \to 0 \neq |p| \leqslant \ell'$ & $|p| \equiv 1 \,(\mathrm{mod}\, 1+K)$ & $|\lambda| \leqslant \ell$.

Assume the RHS of (A$'$.0) in the following. Also let q and J denote the numbers defined above.

(A$'$.1) The case $|p| = \ell'$:

$$A'(\bar{x},\lambda,p) \leftrightarrow B(\bar{x},\lambda+\#(J),q).$$

(A$'$.2) The case $|p| < \ell'$:

$$A'(\bar{x},\lambda,p) \leftrightarrow I(\bar{D}(\bar{x},\lambda+\#(J),q), \{A'(\bar{x},\lambda,pi0^{[K-j]}1^{[j]}):\ i<2\ \&\ j \leqslant K\}).$$

In the RHS $A'(\bar{x},\lambda,pi0^{[K-j]}1^{[j]})$ is substituted for $A(\bar{x},\lambda+j,pi)$ in (A.2). The definition is based on induction along a $2(1+K)$-branching tree, cf. Lemma 2.3.

Claim 2.2. *For a given* p *with* $|p| \equiv 1 \,(\mathrm{mod}\, 1+K)$, *let* q *and* J *denote the numbers defined above. Then* $A(\bar{x}, \lambda + \#(J), q) \Leftrightarrow_{\mathrm{df}} A'(\bar{x}, \lambda, p)$ *enjoys* (A.1) *and* (A.2) *stated in Lemma* 2.5 *provided that* $\#(J)$ *has been defined so that* (8) *holds.*

Proof of Claim 2.2.
(A.1) By (A$'$.1) we have for $|p| = \ell' = 1 + (\ell - 1)(1 + K)$, $A(\bar{x}, \lambda + \#(J), q) \leftrightarrow A'(\bar{x}, \lambda, p) \leftrightarrow B(\bar{x}, \lambda + \#(J), q)$.
(A.2) For each $j \leqslant K$, let $J_j = J * (10^{[K-j]} 1^{[j]})$, i.e., $|J_j| = |J| + K$ and $MSP(J_j, K) = J$ & $\forall i < K(Bit(i, J_j) = 1 \leftrightarrow i < j)$. Then qi and J_j are the numbers defined from $pi0^{[K-j]} 1^{[j]}$. We see $\#(J_j) = \#(J) + j$ from (8). By (A$'$.1) we have for $|p| < \ell'$ and $\bar{D}(q) \Leftrightarrow_{\mathrm{df}} \bar{D}(\bar{x}, \lambda + \#(J), q)$,

$$\begin{aligned} A(\bar{x}, \lambda + \#(J), q) &\leftrightarrow A'(\bar{x}, \lambda, p) \\ &\leftrightarrow I(\bar{D}(q), \{A'(\bar{x}, \lambda, pi0^{[K-j]}1^{[j]}):\ i < 2\ \&\ j \leqslant K\}) \\ &\leftrightarrow I(\bar{D}(q), \{A(\bar{x}, \lambda + \#(J_j), qi):\ i < 2\ \&\ j \leqslant K\}) \\ &\leftrightarrow I(\bar{D}(q), \{A(\bar{x}, \lambda + \#(J) + j, qi):\ i < 2\ \&\ j \leqslant K\}). \end{aligned}$$

This shows Claim 2.2.

Conversely for a given q with $|q| = 1 + r$, let p denote the number such that $|p| = 1 + r(1 + K)$, and $Bit(i, p) = 1 \leftrightarrow \exists j\, [Bit(j, q) = 1\ \&\ i = K + j(1 + K)]$ for $i < |p| \dot{-} 1$. Then $A(\bar{x}, \lambda, q) \Leftrightarrow_{\mathrm{df}} A'(\bar{x}, \lambda, p)$ enjoys (A.1) and (A.2) by Claim 2.2 since $\#(J) = 0$ for the number J defined from the p.

Thus the problem reduces to showing the following proposition, cf. Lemma 2.4:

Proposition 2.1. $\#(J)$ *is* Σ^{b}_0*-bitdefinable for* $|J| \leqslant \ell \|\bar{x}\|$ *so that* (8) *holds.*

This is a bounded counting with a bound $\ell \|\bar{x}\|$, while Corollary 2.1 is a bounded counting with a bound $|x|$. By repeating the above proofs Proposition 2.1 reduces to show the following.

Proposition 2.2. $\#(J)$ *is* Σ^{b}_0*-definable for* $|J| \leqslant c|x|_3 + c$ *and any constant* c *so that* (8) *holds.*

Proof of Proposition 2.2. Suppose $|J| \leqslant c\,|x|_3 + c$. It suffices to show that $y \leqslant \#(J)$ is Σ^{b}_0-definable. Define $y \leqslant \#(J)$ by the formula

$$\exists |u| \leqslant \|J\| \cdot y + 1 \forall i < y[Bit(u_i, J) = 1\ \&\ (i + 1 < y \to u_i < u_{i+1})\ \&\ y \leqslant |J|],$$

where $u_i = u[\|J\| \cdot i,\ \|J\| \cdot (i + 1))$. This means that for some u_i, $u = u'_{y-1} * \cdots * u'_1 * u'_0$ for $u'_i = 2^{\|J\|} + u_i$ and $u_0 < u_1 < \cdots < u_{y-1} < |J|$ & $\{u_i\!:\ i < y\} \subseteq \{v < |J|\!:\ Bit(v, J) = 1\}$. Each $u_i < |J|$ and hence $\|J\|$-digits suffices to represent u_i in binary notation. Also note that any multiplication occurring in this definition is a multiplication for small numbers, cf. $multi(i, j, x, y)$ in Section 1.

The quantifier $\exists |u| \leqslant \|J\| \cdot y$ with $y \leqslant |J|$ is a sharply bounded one since $\|J\| \cdot |J| \leqslant \ell \|x\|$ for some linear form ℓ by the supposition $|J| \leqslant c|x|_3 + c$.

Now define $y = \#(J)$ by $y \leqslant \#(J)$ & $\neg(y+1 \leqslant \#(J))$. Then this is equivalent to $\{u_i\colon i<y\} = \{v<|J|\colon Bit(v,J)=1\}$ for some $u = u'_{y-1} * \cdots * u'_1 * u'_0$ with $u'_i = 2^{\|J\|} + u_i$ and $u_0 < u_1 < \cdots < u_{y-1} < |J|$. From this we see that (8) holds. $\square$

From Theorem 2.1 we see that the multiplication is Σ^b_0-bitdefinable in AID: $x \cdot y = \sum_{i<|y|} f(i,x,y)$ for $f(i,x,y) = x \cdot 2^i \cdot Bit(i,y)$ with $y = \sum_{i<|y|} 2^i \cdot Bit(i,y)$.

3. *ALOGTIME* are Σ^b_0-definable in *AID*

In this section we show that each predicate in *ALOGTIME* is Σ^b_0-definable in *AID*.

Theorem 3.1. *Each predicate in ALOGTIME is Σ^b_0-definable in AID.*

Proof. (cf. Belcarav et al. [2, p. 64, p. 77] for indexing alternating Turing machines). Let A be a predicate in *ALOGTIME* and $M = (Q, \Sigma, \delta, q_0, g)$ be an alternating Turing machine which recognizes A such that M always halts in time $\ell\|x\|$ on input x for a linear form ℓ. We assume that:

1. Q is a finite set of states and $q_0 \in Q$ is an initial state.
2. $\Sigma = \{0,1\}$, $\bar{\Sigma} = \{0,1,b\}$, where b denotes the blank.
3. $g: Q \to \{\wedge, \vee, accept, reject\}$.
4. δ is a transition function such that $\delta: Q \times \bar{\Sigma}^{k+2} \to \mathscr{P}(\bar{\Sigma}^{k+1} \times H^{k+1} \times Q)$ with $H = \{L, N, R\}$.
5. The meanings of L, R, N are given as follows. L: moving one cell to the left, R: moving one cell to the right, N: do not move.
6. M contains $(k+2)$-tapes: a read-only input tape, k-work tapes and an index tape. M writes down a number $i \leqslant |x|$ for an input x in binary notation on the index tape to read the ith input symbol on the input tape. These tapes are numbered in this order. Thus the input tape is referred to as the 0th tape.

Without loss of generality, we can assume that the computation tree of M on input x is a binary tree of depth $\ell\|x\|$. Each $w \in \Sigma^*$ corresponds to a node in a computation tree. For a $w \in \Sigma^*$ let $pd(w) \in \Sigma^*$, denote a word such that $w = pd(w)j$ for some $j \in \Sigma$, i.e., $pd(w)$ is obtained from w by deleting the rightmost symbol in Σ. Put $w_0 \subset w_1 \Leftrightarrow_{\rm df} w_0$ is an initial segment of w_1 for words w_0, w_1.

Thus, we have functions (definable by some terms in the language L_{BA}) $\delta^\Sigma, \delta^H, \delta^Q$ such that for $q \in Q$, $\bar{s} \in \bar{\Sigma}^{k+2}$, $1 \leqslant j \leqslant k+1$, $w \in \Sigma^*$ with $|w| \leqslant \ell\|x\|$

$$(\delta^\Sigma(q,\bar{s},j,w)\colon\ 1 \leqslant j \leqslant k+1) * (\delta^H(q,\bar{s},j,w)\colon\ 1 \leqslant j \leqslant k+1) * (\delta^Q(q,\bar{s},w))$$

is in $\delta(q,\bar{s})$.

This $(2k+3)$-tuple denotes the next move at w when at the predecessor node $pd(w)$, the state is q and the scanned symbols are $\bar{s}$. $\delta^\Sigma(q,\bar{s},k+1,w)$ is the symbol written on the index tape.

Let $\bar{q}, \bar{H}, \bar{s}$ denote the following objects:
1. $\bar{q} = (q^i\colon\ i \leqslant \ell\|x\|) \in Q^*,\ q^i \in Q$.
2. $\bar{H} = H_1, \ldots, H_{k+1}$ and for each j with $1 \leqslant j \leqslant k+1$,
   $H_j = (H_j^i\colon\ i \leqslant \ell\|x\|)$ & $H_j^i \in \{L, N, R\}$.
3. $\bar{s} = s_0, s_1, \ldots, s_{k+1}$ and for each $j \leqslant k+1$,
   $s_j = (s_j^i\colon\ i \leqslant \ell\|x\|)$ & $s_j^i \in \{0, 1, b\}$.

Let $I \in \{0,1\}^*$ denote a path through the computation tree, $|I| = \ell\|x\|$. Then these objects denote guesses on I:
1. q^i is a guess of the state at node $w \subset I$ with $|w| = i$.
2. H_j is a guess of the moves of the jth head on I.
3. s_j is a guess of the scanned symbols by the jth head on I.

For j with $1 \leqslant j \leqslant k+1$, $|w| \leqslant \ell\|x\|$ and $\lambda \leqslant |x|$, $C_j(x, \bar{q}, \bar{H}, \bar{s}, \lambda, w)$ denotes the symbol in $\bar{\Sigma}$ written on the λth cell of jth tape at node w when $\bar{q}, \bar{H}, \bar{s}$ are guesses on the path I: First for the empty word ε, $C_j(x, \bar{q}, \bar{H}, \bar{s}, \lambda, \varepsilon) = b$. Next suppose $w \neq \varepsilon$. Let $Position(H_j, w)$ denote the position of jth head at node w. $Position(H_j, w)$ is determined by counting the numbers of L's and R's in the first $|w|$ part of H_j. Thus $Position(H_j, w)$ is Σ^{b}_0-definable.

Case 1: $\forall w_1 \subset w(\lambda \neq Position(H_j, w_1))$: λ is not and has not been the position where jth head stays at node w or has stayed before w. Then $C_j(x, \bar{q}, \bar{H}, \bar{s}, \lambda, i) = b$.

Case 2: $\exists w_1 \subset w(\lambda = Position(H_j, w_1))$: let w_1 denote the latest such node. Letting $w_0 = pd(w_1)$, $C_j(x, \bar{q}, \bar{H}, \bar{s}, \lambda, w) = \delta^{\Sigma}(q^{|w_0|}, \bar{s}^{|w_0|}, j, w_1)$ with $\bar{s}^{|w_0|} = (s_j^{|w_0|}\colon\ 0 \leqslant j \leqslant k+1)$.

Thus $C_j(x, \bar{q}, \bar{H}, \bar{s}, \lambda, i)$ is Σ^{b}_0-definable in L_{AID}.

Letting $w_0 = pd(w)$ put

$$Init(\bar{q}, \bar{H}, \bar{s}) \ \Leftrightarrow_{\mathrm{df}}\ q^0 = q_0(\text{initial state}) \ \&\ \bigwedge \{s_j^0 = b\colon\ j \leqslant k+1\}.$$

$$State(\bar{q}, \bar{s}, I) \ \Leftrightarrow_{\mathrm{df}}\ \forall w \subset I[w \neq \varepsilon \Rightarrow q^{|w|} = \delta^{Q}(q^{|w_0|}, \bar{s}^{|w_0|}, w)].$$

$$Head(\bar{q}, \bar{H}, \bar{s}, I) \ \Leftrightarrow_{\mathrm{df}}\ \forall w \subset I[w \neq \varepsilon \Rightarrow \bigwedge_{1 \leqslant j \leqslant k+1} H_j^{|w|} = \delta^{H}(q^{|w_0|}, \bar{s}^{|w_0|}, j, w)].$$

$$Symbol(\bar{q}, \bar{H}, \bar{s}, I) \Leftrightarrow_{\mathrm{df}} \forall w \subset I \left[w \neq \varepsilon \Rightarrow s_0^{|w|} = Bit(C_{k+1}^{w}, x)\ \& \right.$$
$$\left. \bigwedge_{1 \leqslant j \leqslant k+1} s_j^{|w|} = C_j(x, \bar{q}, \bar{H}, \bar{s}, \lambda, w) \right],$$

where $\lambda = Position(H_j, w)$ and C_{k+1}^{w} denotes a number in binary notation, i.e., $C_{k+1}^{w} \in \{0,1\}^*$ such that $Bit(\lambda, C_{k+1}^{w}) = 1 \Leftrightarrow C_{k+1}(x, \bar{q}, \bar{H}, \bar{s}, \lambda, w) = 1$, in other words C_{k+1}^{w} is the content of the $(k+1)$-th index tape at the node w.

Let $q(w) = q(w, I)$ denote the state at a node $w \subset I$:

$$q(w) = q \ \Leftrightarrow\ \exists \bar{q}\, \exists \bar{H}\, \exists \bar{s}[Init(\bar{q}, \bar{H}, \bar{s}) \ \&\ State(\bar{q}, \bar{s}, I) \ \&\ Head(\bar{q}, \bar{H}, \bar{s}, I)$$
$$\&\ Symbol(\bar{q}, \bar{H}, \bar{s}, I) \ \&\ q^{|w|} = q],$$

$\bar{q}, \bar{H}, \bar{s}$ are words on length at most $\ell'\|x\|$ for a linear ℓ' over finite alphabets $Q, H = \{L, N, R\}$, $\bar{\Sigma} = \{0, 1, b\}$, resp. Therefore these existential quantifiers are sharply bounded and hence $q(w)$ is Σ_0^{b}-definable. Further existence and uniqueness conditions for $\bar{q}, \bar{H}, \bar{s}$ are provable from Σ_0^{b}-LIND.

Define a Σ_0^{b}-predicate $A_{\mathrm{M}}(x,w)$ in L_{AID} such that

$$|w| = \ell\|x\| \rightarrow [A_{\mathrm{M}}(x,w) \leftrightarrow q(w) \text{ is an accepting state, i.e., } g(q(w)) = accept]$$

and

$$\begin{aligned}|w| < \ell\|x\| \rightarrow [&A_{\mathrm{M}}(x,w)\\ &\leftrightarrow [q(w) \text{ is a universal state, i.e., } g(q(w)) = \wedge \ \& \ A_{\mathrm{M}}(x,w0) \ \& \ A_{\mathrm{M}}(x,w1)] \vee \\ &[q(w) \text{ is an existential state, i.e., } g(q(w)) = \vee \ \& \ (A_{\mathrm{M}}(x,w0) \vee A_{\mathrm{M}}(x,w1))]].\end{aligned}$$

Thus the given predicate $A \in ALOGTIME$ is defined by $A(x) \leftrightarrow A_{\mathrm{M}}(x,\varepsilon)$. □

Remark. If we do not guess $\bar{q}, \bar{H}, \bar{s}$ and try to define directly the configuration $C(x,i)$ on input x at the node i, then the resulting definition would involve a complicated simultaneous inductive definition, even if M is a deterministic Turing machine with run times at most $\ell\|x\|$.

4. Consistency proof of Frege system

In this section we show that a truth definition for PLOF formulae (Postfix-Longer-Operands-First) is Σ_0^{b}-definable in *AID*. This is done by mimicking the proofs in Buss [7] almost word for word. The reader is recommended to have a copy of [7] in hand. Although one could apply the simplified algorithm for boolean formula evaluation in [9], we stick to [7], since the latter gave a full proof of the fact that the truth definition respects the meanings of propositional connectives.

In the next section we show that, if $f(\bar{x})$ is a Σ_0^{b}-bitdefinable function in *AID*, then $C(f(\bar{x}))$ is Σ_0^{b}-definable in *AID* for any Σ_0^{b}-formula $C(y)$, cf. Lemma 5.6. Therefore we can use freely such functions in Σ_0^{b}-formulae.

Let x be (a code of) a sequence of $19 < 2^5$ symbols in the set
$\Sigma = \{p, 0, 1, (,)$ (parentheses), (comma), 13 propositional connectives$\}$,
where propositional connectives are unary or binary, cf. [7, p. 8].

1. the length $|x|_\Sigma$ of x as a word from Σ, $5|x|_\Sigma = |x|$.
2. jth symbol from Σ in x: $Sym_j^x = x[5(j-1), 5j)$ for $1 \leqslant j \leqslant |x|_\Sigma$.
3. A *logical symbol* is a parenthesis, comma, propositional connective or propositional variable p_i (i is a word on $\{0,1\}$).
4. $R(x,j,i) \Leftrightarrow_{\mathrm{df}} i = \#\{k \leqslant j : Sym_k^x$ is not 0 nor 1$\}$.
   Let 'Sym_j^x is in $x[i]$' $\Leftrightarrow_{\mathrm{df}} R(x,j,i)$.
5. $x[i]$ = the ith logical symbol of $x = x[5m, 5n)$, where
   $m = \min\{j \leqslant |x| : Sym_j^x$ is in $x[i]\}$,
   $n = \max\{j \leqslant |x| : Sym_j^x$ is in $x[i]\} + 1$.

Note that $\min\{j \leqslant |x|: \cdots\}$, $\max\{j \leqslant |x|: \cdots\}$ with Σ^b_0 conditions $\cdots$ are available in Σ^b_0-formulae by Σ^b_0-LIND.

$$|x|_{\mathrm{L}} = \text{the number of logical symbols in } x = \#\{j \leqslant |x|: \ Sym^x_j \text{ is not } 0 \text{ nor } 1\}.$$

First of all we have to develop metamathematics, arithmetization of syntax, e.g., define x to be (a code of) a postfix formula iff $x[1]$ is an atomic formula, the number $\#i \leqslant |x|_{\mathrm{L}}$ ($x[i]$ is an atomic formula) is equal to $1 + \#i \leqslant |x|_{\mathrm{L}}$ ($x[i]$ is a binary connective) and the number $\#i \leqslant j(x[i]$ is an atomic formula) is larger than the number $\#i \leqslant j(x[i]$ is a binary connective) for any $j < |x|_{\mathrm{L}}$.

This requires *countings*, $C_\varphi(x) = \#\{i < |x|: \ \varphi(i)\}$. It is easy to see the

Proposition 4.1. *There exists a Σ^b_0-formula $PL(x,i)$ such that if x is an infix formula, then $y = \{i < |x|: \ PL(x,i)\}$ is the PLOF form of x, i.e., for any $k < |x|_{\mathrm{L}}$, the kth symbol of y is the kth PLOF symbol of x, cf.* [7, p. 12].

In the following, otherwise stated, x denotes a PLOF formula. By Σ^b_0-LIND, which corresponds to brute forth induction in [7], we have, cf. [7, p. 16]: $\forall j \leqslant |x|_L \exists! i \leqslant |x|_L \{x[i,j]$ is a formula$\}$, where $x[i,j]$ denotes a substring of x from $x[i]$ through $x[j]$ inclusive. Therefore, we have a Σ^b_0-definable function x_j such that

$$x_j = \text{the unique subformula of } x \text{ of the form } x[i,j].$$

For $i,j \leqslant |x|_L$, cf. [7, p. 16],

$$j \trianglelefteq i \ \Leftrightarrow_{\mathrm{df}} \ x[j] \text{ is in } x_i \Leftrightarrow_{\mathrm{df}} k \leqslant j \leqslant i \text{ with } x_i = x[k,i]$$

$$lca(j,i) =_{\mathrm{df}} \min\{k \leqslant |x|_L: \ i \trianglelefteq k \ \& \ j \trianglelefteq k\}.$$

Let $x[i,j]$ be a $\leqslant 1$-scarred formula, cf. [7, p. 15]. Suppose $l < r$, $l < j$ and $i \leqslant r$. Then k is the *breakpoint of* $x[i,j]$ 1-*selected by* $(l,r]$ if (cf. [7, p. 16]).

$k = \max\{k \leqslant \min\{r,j\}$: either $x[l+1]$ or $x[i]$ is in $x_k\}$. This is a Σ^b_0-definition.

Define, cf. [7, p. 16], Δ_u and ε_u inductively: $\Delta_0 = 2$, $\varepsilon_u = \lfloor \frac{1}{2}\Delta_u \rfloor$, $\Delta_{u+1} = \Delta_u + \varepsilon_u$, i.e., $\Delta_u = \lfloor \frac{3}{2} \lfloor \frac{3}{2} \cdots \lfloor \frac{3}{2} 2 \rfloor \cdots \rfloor \rfloor$ with u's $\lfloor \frac{3}{2} \cdot \rfloor$.

Δ_u is needed to be defined up to $|x|_{\mathrm{L}} < \Delta_{u-1}$. Therefore $u < \|x\|$ suffices.

To see that $\Delta_u \, (u < \|x\|)$ is Σ^b_0-definable, we use the carry–save–addition as in the proof of Theorem 2.1: for $|p| \leqslant \|x\|$, the node p codes the number $\{S(x,\lambda,p) + C(x,\lambda,p)\}_\lambda$, and the number corresponds to $\Delta_{\|x\|-|p|}$ if p is even, and to $\varepsilon_{\|x\|-|p|}$ if p is odd. For $u < \|x\|$, $\Delta_u < (\frac{3}{2})^{u+2} < 2^{u+2} \leqslant 8|x|$. Thus we can use the functions Δ_u and ε_u in Σ^b_0-formulae.

One can Σ^b_0-define the following in [7, pp. 17–19]:

1. *Breakpoints* a_p $(p = 1,2,3,4)$ *of* $x[i,j]$ *generated by* $(m,n]$ with $n - m = \Delta_{u+1}$ for some $u \geqslant 0$.
   (a) a_1 is the breakpoint of $x[i,j]$ 1-selected by $(m, m+\varepsilon_u]$
   (b) a_2 is the breakpoint of $x[i,j]$ 1-selected by $(m+\varepsilon_u, n-\varepsilon_u]$

(c) a_4 is the least common ancestor $lca(a_1, a_2)$ of a_1, a_2
(d) $a_3 = a_4 - 1$.

2. For a formula $x[i, j]$ and numbers n, m such that $m < i \leqslant j \leqslant n$ and $n - m = \Delta_{u+1}$, split the formula $x[i, j]$ into up to $\leqslant 4$ subformulae $SubFm_1(x, [i, j], (m, n], p)\,(p = 1, 2, 3, 4)$ by introducing the breakpoints a_p generated by $(m, n]$:

$$SubFm_1(x, [i, j], (m, n], 1) = [i, a_1],$$

$$SubFm_1(x, [i, j], (m, n], 2) = [a_1 + 1, a_2],$$

$$SubFm_1(x, [i, j], (m, n], 3) = [a_2 + 1, a_3],$$

$$SubFm_1(x, [i, j], (m, n], 4) = [a_4 + 1, j],$$

where breakpoints are defined so that, if $a_2 \neq a_4$, then $Bin\mathrm{Op}(x, [i, j], (m, n]) = x[a_4]$ is a binary connective.

3. $x[i]$ is a *scar* of the interval $[a, b]$ iff $i < a$ and there is a connective $x[k]$ with $a \leqslant k \leqslant b$ such that x_i is one of the operands of $x[k]$.

Then Lemma 12 in [7, p. 19] is provable in *AID*.

Lemma 4.1 (Buss, [7, Lemma 12 in p. 19]). *AID proves the followings: let $x[i, j]$ be a $\leqslant 1$-scarred formula, $n - m = \Delta_{u+1}$ and a_p $(p = 1, 2, 3, 4)$ the breakpoints of $x[i, j]$ generated by $(m, n]$.*

(a) $i \leqslant m + \varepsilon_u < j \Rightarrow a_1 + 1 \trianglelefteq a_2$,
(b) $\max\{m + 1, i\} \leqslant a \leqslant a_2 \Rightarrow a \trianglelefteq a_1 \vee a \trianglelefteq a_2$, *and* $\max\{m + 1, i\} \leqslant a \leqslant a_3 \Rightarrow a \trianglelefteq a_1 \vee a \trianglelefteq a_2 \vee a \trianglelefteq a_3$.
(c) *For* $p = 1, 2, 3, 4$, $SubFm_1(x, [i, j], (m, n], p)$ *does not have more than one scar* $x[k]$ *with* $k \geqslant \max\{m + 1, i\}$

In the proof of Lemma 4.1(a) use the fact that x is a PLOF formula. For a proof of Lemma 4.1 we need:

Proposition 4.2. 1. $a \trianglelefteq b \Rightarrow a \leqslant b$
2. $x_b = x[a, b] \Rightarrow (a \leqslant c \leqslant b \Leftrightarrow c \trianglelefteq b)$,
3. $a < b$ & $a \perp b (\Leftrightarrow_{\mathrm{df}} \neg(a \trianglelefteq b \vee b \trianglelefteq a))$ & $c = lca(a, b) \Rightarrow b \trianglelefteq c - 1$.

cf. [7, p. 21]. Let $n - m = \Delta_{u+1}$. For $k \leqslant u + 1$ and $1 \leqslant p_1, \ldots, p_k \leqslant 4$, (the sequence $p_1, \ldots, p_k$ is coded by a number of length $1 + 2k \leqslant 3 + 2u < 3 + 2\|x\|$) define

$$Int_k((m, n], p_1, \ldots, p_k) = (m', n']$$

with

$$m' = m + \sum_{j=1}^{k} \left\lfloor \frac{1}{2}(p_j - 1) \right\rfloor \cdot \varepsilon_{u+1-j} \quad \text{and} \quad n' = m' + \Delta_{u+1-k}$$

and

$$\lfloor \tfrac{1}{2}(p-1) \rfloor = \begin{cases} 0 & \text{if } p = 1, 2, \\ 1 & \text{if } p = 3, 4. \end{cases}$$

This can be Σ_0^{b}-defined by using *vector summation* $g(x, \bar{y}) = \sum_{i<|x|} f(i, \bar{y})$.

Definition 4.1. Iteration of spliting into subformulae SubFm_k. cf. [7, p. 22]. For $k \leqslant u+1$, $1 \leqslant p_1, \dots, p_k \leqslant 4$, $1 \leqslant l \leqslant k$ let $a_1^l, \dots, a_4^l$ be breakpoints of $x[i,j]$ generated by $\mathit{Int}_{l-1}((m,n], p_1, \dots, p_{l-1})$. Put $a_0^l = i-1$, $a_5^l = j$ and

$$i_l = \begin{cases} p_l - 1 & \text{if } 1 \leqslant p_l \leqslant 3, \\ 4 & \text{if } p_l = 4. \end{cases}$$

Then $\mathit{SubFm}_1(x, [i,j], \mathit{Int}_{l-1}((m,n], p_1, \dots, p_{l-1}), p_l) = [a_{i_l}^l + 1, a_{1+i_l}^l]$. Put

$$c_k = \max\{a_{i_l}^l \colon 1 \leqslant l \leqslant k\}, \quad d_k = \min\{a_{1+i_l}^l \colon 1 \leqslant l \leqslant k\}.$$

Define for $(m', n'] = \mathit{Int}_{k-1}((m,n], p_1, \dots, p_{k-1})$

$$\begin{aligned} \mathit{SubFm}_k(x, [i,j], (m,n], p_1, \dots, p_k) &=_{\mathrm{df}} [c_k + 1, d_k] \\ &= \bigcap_{1 \leqslant l \leqslant k} \mathit{SubFm}_1(x, [i,j], \mathit{Int}_{l-1}((m,n], \\ &\quad\ p_1, \dots, p_{l-1}), p_l) \\ &= \mathit{SubFm}_{k-1}(x, [i,j], (m,n], p_1, \dots, p_{k-1}) \\ &\quad\ \cap \mathit{SubFm}_1(x, [i,j], (m', n'], p_k). \end{aligned}$$

Lemma 4.2 (Buss [7, Lemma 13 in p. 22]). *AID proves the following: Suppose* $x[i,j]$ *is a* $\leqslant 1$*-scarred subformula,* $n < i \leqslant j \leqslant m$, $n - m = \Delta_{u+1}$, $k \geqslant 0$ *and* $1 \leqslant p_1, \dots, p_k \leqslant 4$, *and let A denote the interval* $\mathit{SubFm}_k(x, [i,j], (m,n], p_1, \dots, p_k)$. *Then*

(a) *A is properly contained in* $\mathit{Int}_k((m,n], p_1, \dots, p_k)$.

(b) *Each symbol in A is in exactly one of the intervals* $\mathit{SubFm}_{k+1}(x, [i,j], (m,n], p_1, \dots, p_k, p_{k+1})$ $(1 \leqslant p_{k+1} \leqslant 4)$ *or is the binary operator* $\mathit{Bin}\mathrm{Op}(x, A, \mathit{Int}_k((m,n], p_1, \dots, p_k))$.

(c) *Each* $\mathit{SubFm}_{k+1}(x, [i,j], (m,n], p_1, \dots, p_k, p_{k+1})$ $(1 \leqslant p_{k+1} \leqslant 4)$ *is a* $\leqslant 1$*-scarred subformula.*

Finally define the truth value of $x[i,j]$ by synthesizing truth values $\mathit{Value}_k(x, [i,j], (m,n], \boldsymbol{p})$ of subformulae $\mathit{SubFm}_k(x, [i,j], (m,n], \boldsymbol{p})$ $(\boldsymbol{p} = p_1, \dots, p_k)$.

Definition 4.2 (*Buss* [7, p. 23]). Let $n - m = \Delta_{u+1}$, $m < i \leqslant j \leqslant n$, $x[i,j]$ is a $\leqslant 1$-scarred formula, $0 \leqslant k \leqslant u+1, 1 \leqslant p_1, \dots, p_k \leqslant 4$.

$\mathit{Value}_k(x, [i,j], (m,n], \boldsymbol{p})$ $(\boldsymbol{p} = p_1, \dots, p_k)$ is defined by

Case 1: $k = u+1$: Then $\mathit{Int}_k(x, (m,n], \boldsymbol{p}) = (a, a+2]$ for some a.

If $\mathit{SubFm}_k(x, [i,j], (m,n], \boldsymbol{p})$ is undefined, then

$$\mathit{Value}_k(x, [i,j], (m,n], \boldsymbol{p}) =_{\mathrm{df}} (\top, \bot).$$

Otherwise $SubFm_k(x,[i,j],(m,n],\boldsymbol{p})$ consists of a single logical symbol, $\neg$ or $\top$ or $\bot$ or a variable q. Then $Value_k(x,[i,j],(m,n],\boldsymbol{p})$ is defined to be $(\bot,\top)$ or $(\top,\top)$ or $(\bot,\bot)$ or (q,q), resp. By (q,q) we mean $(\top,\top)$ if q has truth value *True* and $(\bot,\bot)$ if q has truth value *False*.

Cases 2 and 3: $k \leqslant u$: Let $(a,b] = Int_k(x,[i,j],(m,n],\boldsymbol{p}), 1 \leqslant p_{k+1} \leqslant 4$. Put $I_{p_{k+1}} = SubFm_{k+1}(x,[i,j],(m,n],\boldsymbol{p},p_{k+1})$. Then by Lemma 4.2(b) $A = SubFm_k(x,[i,j],(m,n],\boldsymbol{p}) = I_1 \dot\cup I_2 \dot\cup I_3 \dot\cup I_4 \dot\cup Bin\mathrm{Op}(x,A,Int_k((m,n],\boldsymbol{p}))$ (disjoint union) by using break-points $a_1,\dots,a_4$ of $x[i,j]$ generated by $(a,b]$. Let $v_p\,(p=1,2,3,4)$ denote the truth value

$$v_p = Value_{k+1}(x,[i,j],(m,n],\boldsymbol{p},p).$$

Case 2: $a_2 = a_4$: Then define

$$Value_k(x,[i,j],(m,n],\boldsymbol{p}) = v_1 \circ v_2 \circ v_4$$

for the (reverse) composition $\circ$:

$$(r_1,r_2) = (s_1,s_2) \circ (t_1,t_2) \Leftrightarrow_{\mathrm{df}} r_i = \begin{cases} t_1 & \text{if } s_i = \top, \\ t_2 & \text{if } s_i = \bot. \end{cases}$$

Case 3: $a_2 \neq a_4$: Then define

$$Value_k(x,[i,j],(m,n],\boldsymbol{p}) = f_{Bin\mathrm{Op}}(v_1,v_2) \circ v_3 \circ v_4,$$

where, if $Bin\mathrm{Op}(x,[i,j],Int_k((m,n],\boldsymbol{p})) = \odot$, then

$$f_{Bin\mathrm{Op}}((s_1,s_2),(t_1,t_2)) =_{\mathrm{df}} (s_1 \odot t_1, s_2 \odot t_2).$$

(In this case we have $t_1 = t_2$.)

Now, we examine the definition of $Value_k(x,[i,j],(m,n],p_1,\dots,p_k)$ in *AID*. Let $Value_k^{\xi}(x,[i,j],(m,n],\boldsymbol{p})\,(\boldsymbol{p} = p_1,\dots,p_k)$ be a predicate for $\xi \in \{\top,\bot\}$ such that if

$$Value_k(x,[i,j],(m,n],\boldsymbol{p}) = (\xi_1,\xi_2)\,(\xi_1,\xi_2 \in \{\top,\bot\}),$$

then

$$\begin{aligned} &Value_k^{\top}(x,[i,j],(m,n],\boldsymbol{p}) \text{ holds } \Leftrightarrow \xi_1 = \top, \\ &Value_k^{\bot}(x,[i,j],(m,n],\boldsymbol{p}) \text{ holds } \Leftrightarrow \xi_2 = \top. \end{aligned}$$

Definition 4.2 gives a *simultaneous inductive definition of the predicates* $Value_k^{\xi}(x,[i,j],(m,n],\boldsymbol{p})\,(\xi \in \{\top,\bot\})$ *along a quadtree of depth* $u+2 < \|j-i\|+2 \leqslant ||x|_\Sigma| + 2 \leqslant \|x\|+2$: for some Σ_0^{b} $B,\bar{D}$ in L_{AID} and a boolean I.

Case 1: $k = u+1$: $Value_k^{\xi}(x,[i,j],(m,n],\boldsymbol{p}) \leftrightarrow B(x,i,j,m,n,\boldsymbol{p},\xi)$.

Cases 2 and 3: $k \leqslant u$: $Value_k^{\xi}(x,[i,j],(m,n],\boldsymbol{p})$ iff $I(\bar{D}(x,i,j,m,n,\boldsymbol{p},\xi),$ $\{Value_{k+1}^{\eta}(x,[i,j],(m,n],\boldsymbol{p},i)\colon\ i=1,\dots,4,\ \ \eta \in \{\top,\bot\}\})$ ($Bin\mathrm{Op}(x,[i,j],Int_k((m,n],\boldsymbol{p}))$ is one of the finitely many binary connectives, and so can be written as a finite disjunction.)

Thus by Lemmata 2.3, 2.4 and 2.5, $Value_k^{\xi}(x,[i,j],(m,n],\boldsymbol{p})$ is Σ_0^{b}-definable in *AID*.

Further this truth definition respects the meanings of propositional connectives and the truth value $Value_0(x,[i,j],(m,n])$ is independent of m and n.

Assume that $x[i,j]$ is a $\leqslant 1$-scarred formula, $n-m=\Delta_{u+1}, m<i\leqslant j\leqslant n$. Then we have the following lemmata and corollary by Σ_0^{b}-LIND.

Lemma 4.3 (Buss [7, Lemma 14 in p. 24]). *If $x[j]$ is a unary connective $\odot$, then $Value_0(x,[i,j],(m,n])=Value_0(x,[i,j-1],(m,n])\circ(s_1,s_2)$ where (s_1,s_2) is the pair of boolean truth values giving the truth value of the $\leqslant 1$-scarred formula $\odot$. (If $\odot=\neg$, then $(s_1,s_2)=(\bot,\top)$).*

Lemma 4.4 (Buss [7, Lemma 15 in p. 25]). *Suppose that $x[j]$ is a binary connective $\odot$ and let $f_\odot$ be the binary function such that $f_\odot((s_1,s_2),(t,t))=(s_1\odot t,s_2\odot t)$. Then*
(a) *If $x[i,j-1]$ is an unscarred formula,*

$$Value_0(x,[i,j],(m,n])=f_\odot((\top,\bot),Value_0(x,[i,j-1],(m,n])).$$

(b) *Otherwise, let $k\in[i,j]$ be such that $x[k,j-1]$ is a formula (unscarred). Then*

$$\begin{aligned}&Value_0(x,[i,j],(m,n])\\&\quad=f_\odot(Value_0(x,[i,k-1],(m,n]),Value_0(x,[k,j-1],(m,n])).\end{aligned}$$

Corollary 4.1 (Buss [7, Corollary 16 in p. 25]). *If $x[i,j]$ is a formula and if $m_k<i\leqslant j\leqslant j_k$, $n_k-m_k=\Delta_{u_k+1}$ for $k=1,2$, then*

$$Value_0(x,[i,j],(m_1,n_1])=Value_0(x,[i,j],(m_2,n_2]).$$

Thus we can Σ_0^{b}-define the truth for PLOF formulae by

$$TRUE_{PLOF}(x,[i,j])=Value_0^{\xi}(x,[i,j],(m,n])$$

with $\xi\in\{\top,\bot\}, n-m=\Delta_{u+1}$ and $m<i\leqslant j\leqslant n$, e.g., $m=i-1, u=\|j-i\|-1$.

Let $RFN(PLOF\text{-}\mathscr{F})$ denote

$$\forall x\in PLOF[PLOF\text{-}\mathscr{F}\vdash x\to TRUE_{PLOF}(x)],$$

the reflection schema for a PLOF Frege system. Proofs in $PLOF\text{-}\mathscr{F}$ are sequences of PLOF formulae separated commas. By counting commas we can Σ_0^{b}-define a function $\beta(i,x)=i$th formula of a proof x.

For an infix formula x we set

$$TRUE(x)\ \Leftrightarrow_{\mathrm{df}}\ TRUE_{PLOF}(\{i<|x|:\ PL(x,i)\})$$

for the PLOF form $y=\{i<|x|:\ PL(x,i)\}$ of x, cf. Proposition 4.1, Definitions 5.6, 5.5 and Lemma 5.6. Thus $RFN(\mathscr{F})$ denotes $\forall x[\mathscr{F}\vdash x\to TRUE(x)]$, i.e., $\forall x[\mathscr{F}\vdash x\to TRUE_{PLOF}(\{i<|x|:\ PL(x,i)\})]$, Reflection schema for a Frege system.

By the above examinations of proofs in [7] we conclude the

Theorem 4.1. 1. *AID* $\vdash$ *RFN*(*PLOF*-$\mathscr{F}$) *for any Frege system* $\mathscr{F}$.
2. *AID* $\vdash$ *RFN*($\mathscr{F}$) *for any Frege system* $\mathscr{F}$.

5. Stratifications

In this section *stratifications* of formulae are defined. These are in essence interpretation of first-order formulae as second-order formulae, and to rewrite a formula into an equivalent one in normal form. In later sections we need these.

First we define stratified formulae in L_{BA}.

Definition 5.1 (*Stratified formulae in* L_{BA}). Let $\bar{x}$, u and $\bar{\imath}$ be sequences of variables with $\bar{x}\cap\{u\}\cap\bar{\imath}=\emptyset$. Let $\mathscr{B}_{\mathrm{BA}}(\bar{x};\bar{\imath})$ and $\mathscr{B}_{\mathrm{BA}}(\bar{x};u;\bar{\imath})$ denote the sets of formulae generated as follows:

1. Atomic formulae of the forms

   $$Bit(i,x)=1,\ Bit(i,j)=1,\ p(\bar{\imath})=q(\bar{\imath}),\ p(\bar{\imath})<q(\bar{\imath})$$

   are in $\mathscr{B}_{\mathrm{BA}}(\bar{x};\bar{\imath})\cap\mathscr{B}_{\mathrm{BA}}(\bar{x};u;\bar{\imath})$, where x is in the list $\bar{x}$, i,j are in the list $\bar{\imath}$, $p(\bar{\imath}),q(\bar{\imath})$ are polynomials in $\bar{\imath}$.
2. If $B_0(\bar{x};\bar{\imath}),B_1(\bar{x};\bar{\imath})\in\mathscr{B}_{\mathrm{BA}}(\bar{x};\bar{\imath})$, then $B_0\wedge B_1,B_0\vee B_1,\neg B_0\in\mathscr{B}_{\mathrm{BA}}(\bar{x};\bar{\imath})$. Similarly the set $\mathscr{B}_{\mathrm{BA}}(\bar{x};u;\bar{\imath})$ is closed under propositional connectives.
3. If $B(\bar{x};u;\bar{\imath}^\frown j)\in\mathscr{B}_{\mathrm{BA}}(\bar{x};u;\bar{\imath}^\frown j)$, then $Qj<uB(\bar{x};u;\bar{\imath}^\frown j)\in\mathscr{B}_{\mathrm{BA}}(\bar{x};u;\bar{\imath})$ for $Q\in\{\forall,\exists\}$.
4. If $B(\bar{x};\bar{\imath}^\frown j)\in\mathscr{B}_{\mathrm{BA}}(\bar{x};\bar{\imath}^\frown j)$, then $Qj<p|\bar{x}|B(\bar{x};\bar{\imath}^\frown j)\in\mathscr{B}_{\mathrm{BA}}(\bar{x};\bar{\imath})$ for any polynomial $p|\bar{x}|$ and $Q\in\{\forall,\exists\}$.
5. If $B(\bar{x}^\frown y;\bar{\imath})\in\mathscr{B}_{\mathrm{BA}}(\bar{x}^\frown y;\bar{\imath})$, then $Q|y|\leqslant\ell\|\bar{x}\|B(\bar{x}^\frown y;\bar{\imath})\in\mathscr{B}_{\mathrm{BA}}(\bar{x};\bar{\imath})$ for any linear form $\ell\|\bar{x}\|$ and $Q\in\{\forall,\exists\}$.

We say that a Σ^{b}_0-formula $B(\bar{x};\bar{\imath})$ in L_{BA} is *stratified with respect to* $(\bar{x};\bar{\imath})$ if $B(\bar{x};\bar{\imath})\in\mathscr{B}_{\mathrm{BA}}(\bar{x};\bar{\imath})$. Also we say that a Σ^{b}_0-formula $B(\bar{x})$ in L_{BA} is *stratified with respect to* $\bar{x}$ if $B(\bar{x})\in\mathscr{B}_{\mathrm{BA}}(\bar{x};)$ with the empty list $\bar{\imath}=\emptyset$. When no confusion is likely to occur, we simply write $\mathscr{B}_{\mathrm{BA}}$ for $\mathscr{B}_{\mathrm{BA}}(\bar{x};\bar{\imath})$.

Observe that in a formula in the set $\mathscr{B}_{\mathrm{BA}}(\bar{x};u;\bar{\imath})$, no quantifier $Q|y|\leqslant\ell\|\bar{x}\|$ occurs for a 'second-order' variable y, i.e., a variable in the first list $\bar{x}$. Also note that function 'constants' occurring in a formula in $\mathscr{B}_{\mathrm{BA}}$ are $Bit,+,0,1,\ |x|$ and $|x|\cdot|y|,\ i\cdot j$.

First we exhibit u-stratified forms $B_=(x,y;u),B_<(x,y;u)$ of atomic formulae $x=y$, $x<y$ and $B_f(\bar{x};u;i)$ of the function constants f. These enjoy demonstrably in $\Sigma^{\mathrm{b}}_0-LIND$ the following:

$$\begin{aligned}&u\geqslant\max\{|x|,|y|\}\Rightarrow(B_=(x,y;u)\leftrightarrow x=y)\,\&\,(B_<(x,y;u)\leftrightarrow x<y),\\&u>\sum\{|x_k|\colon x_k\in\bar{x}\}\Rightarrow(B_f(\bar{x};u;i)\leftrightarrow Bit(i,f(\bar{x}))=1).\end{aligned}\tag{9}$$

1. $(=)$: $B_=(x,y;u)\Leftrightarrow_{\mathrm{df}}\forall i<u(Bit(i,x)=1\leftrightarrow Bit(i,y)=1)$.
   $B_=(x,y;u)\leftrightarrow x=y$ if $u\geqslant\max\{|x|,|y|\}$.
2. $(<)$: $B_<(x,y;u)$ iff
   $\exists j<u[Bit(j,x)\neq 1\wedge Bit(j,y)=1\wedge\forall i<u(j<i\rightarrow(Bit(i,x)=1\leftrightarrow Bit(i,y)=1))]$.
   $B_<(x,y;u)\leftrightarrow x<y$ if $u\geqslant\max\{|x|,|y|\}$.

3. (zero): $B_0(;u;i) \Leftrightarrow_{\mathrm{df}} i<i \leftrightarrow Bit(i,0)=1$.
4. (one): $B_1(;u;i) \Leftrightarrow_{\mathrm{df}} i=0 \leftrightarrow Bit(i,1)=1$.
5. (addition): $B_+(x,y;u;i)$ iff
$Bit(i,x)=1 \oplus Bit(i,y)=1 \oplus \exists j<u[j<i \wedge Bit(j,x)=1 \wedge Bit(j,y) = 1 \wedge \forall k<u\{j<k<i \to (Bit(k,x)=1 \oplus Bit(k,y)=1)\}]$,
where $\oplus$ denotes the excluded or. We have $B_+(x,y;u;i) \leftrightarrow Bit(i,x+y)=1$ if $u \geqslant |x|$.
6. (half): $B_{\lfloor x/2 \rfloor}(x;u;i)$ iff
$\exists j<u(Bit(j,x)=1 \wedge j=i+1) \leftrightarrow Bit(i+1,x)=1$.
$B_{\lfloor x/2 \rfloor}(x;u;i) \leftrightarrow Bit(i,\lfloor x/2 \rfloor)=1$ if $u \geqslant |x|$.
7. (modified subtraction): $B_{\dot{-}}(x,y;u;i)$ iff
$B_<(y,x;u) \wedge [\{(Bit(i,x)=1 \oplus Bit(i,y)=1) \wedge \neg B_{LSP}(x,y,i;u) y[0,i) \leqslant x[0,i)\} \vee \{Bit(i,x)=Bit(i,y) \wedge B_{LSP}(x,y,i;u)\}]$,
where $B_{LSP}(x,y,i;u)$ denotes $\exists j<u[j<i \wedge Bit(j,x) \neq 1 \wedge Bit(j,y)=1 \wedge \forall k <u(j<k<i \to (Bit(k,x)=1 \leftrightarrow Bit(k,y)=1))]$.
$B_{LSP}(x,y,i;u) \leftrightarrow x[0,i)<y[0,i)$ if $u \geqslant i$. Hence $B_{\dot{-}}(x,y;u;i) \leftrightarrow Bit(i,x \dot{-} y)=1$ if $u \geqslant \max\{|x|,|y|\}$.
8. (part): $B_{x[y,z)}(x,y,z;u;i)$ iff
$\exists j<u \exists k<u(Bit(j,x)=1 \wedge k+i=j \wedge B_=(k,y;u) \wedge B_<(j,z;u)$.
We have $B_{x[y,z)}(x,y,z;u;i) \leftrightarrow \exists j<|x|(Bit(j,x)=1 \wedge y+i=j<z) \leftrightarrow Bit(i,x[y,z)) =1$ if $u \geqslant \max\{|x|,|z|\}$.
9. (length): $B_{|x|}(x;u;i)$ iff
$\exists j<u[Bit(j,x)=1 \wedge \forall k<u(k>j \to Bit(k,x) \neq 1) \wedge B_+(j,1;u;i)]$.
Since $|x| = \min\{x, \max\{j<|x| : Bit(j,x)=1\}+1\}$, we have
$Bit(i,|x|)=1 \leftrightarrow \exists j<|x|[j=\max\{j<|x| : Bit(j,x)=1\} \wedge Bit(i,j+1)=1]$ and hence $B_{|x|}(x;u;i) \leftrightarrow Bit(i,|x|)=1$ if $u \geqslant |x|$.
10. $(x+1=y): B_{+1}(x,y;u) \Leftrightarrow_{\mathrm{df}} \forall i<u(B_+(x,1;u;i) \leftrightarrow Bit(i,y)=1)$.
$B_{+1}(x,y;u) \leftrightarrow x+1=y$ if $u> \max\{|x|,|y|\}$.
11. $(y=|x|): B_{y=|x|}(y,x;u)$ iff
$\forall i < u(Bit(i,y) \neq 1 \wedge Bit(i,x) \neq 1) \vee \exists j < u[Bit(j,x) = 1 \wedge \forall k<u(k>j \to Bit(k,x) \neq 1) \wedge B_{+1}(j,y;u)]$.
Then $B_{y=|x|}(y,x;u) \leftrightarrow x=y=0 \vee y=\max\{j<|x| : Bit(j,x)=1\} + 1 \leftrightarrow y=|x|$ if $u> \max\{|x|,|y|\}$.
12. (padding): $B_{x \cdot 2^{|y|}}(x,y;u;i)$ iff
$\exists j<u \exists k<u[Bit(j,x)=1 \wedge k+j=i \wedge B_{y=|x|}(k,y;u)]$.
Since we have $Bit(i,x \cdot 2^{|y|})=1 \leftrightarrow \exists j<|x|[Bit(j,x)=1 \wedge i=|y|+j]$, $B_{x \cdot 2^{|y|}}(x,y;u;i) \leftrightarrow Bit(i,x \cdot 2^{|y|})=1$ if $u>|x|+|y|$.
13. (smash): $B_\#(x,y;u;i)$ iff
$\exists j<u \exists k<u(i=j \cdot k \wedge B_{y=|x|}(j,x;u) \wedge B_{y=|x|}(k,y;u))$.
$B_\#(x,y;u;i) \leftrightarrow Bit(i,x\#y)=1$ if $u> \max\{|x|,|y|\}$.

Definition 5.2 (*Bitwise computability*). 1. Let $t(\bar{x})$ be a term with variables $\bar{x}$. (Every variable in $t(\bar{x})$ need not be in the list $\bar{x}$.) We say that $t(\bar{x})$ is *bitwise computable with respect to* $\bar{x}$ denoted by $t(\bar{x}) \in \mathscr{C}_{\mathrm{BA}}$ if there exists a stratified formula

$C_t^*(\bar{x}; i) \in \mathscr{B}_{\mathrm{BA}}(\bar{x}; i)$ such that

$$AID \vdash \forall i < |t(\bar{x})| [Bit(i, t(\bar{x})) = 1 \leftrightarrow C_t^*(\bar{x}; i)]. \tag{10}$$

2. Let $C(\bar{x})$ be a Σ_0^{b}-formula in L_{BA} with variables $\bar{x}$. (Every free variable in $C(\bar{x})$ need not be in the list $\bar{x}$.) We say that $C(\bar{x})$ is *bitwise computable with respect to* $\bar{x}$ denoted by $C(\bar{x}) \in \mathscr{C}_{\mathrm{BA}}$ if there exists a stratified formula $C^*(\bar{x}) \in \mathscr{B}_{\mathrm{BA}}(\bar{x};)$ such that

$$AID \vdash C(\bar{x}) \leftrightarrow C^*(\bar{x}). \tag{11}$$

Lemma 5.1. $t(\bar{x}), s(\bar{x}) \in \mathscr{C}_{\mathrm{BA}} \Rightarrow t(\bar{x}) = s(\bar{x}), t(\bar{x}) < s(\bar{x}) \in \mathscr{C}_{\mathrm{BA}}$.

Proof. Let $p|\bar{x}|$ be a polynomial such that $\max\{|t(\bar{x})|, |s(\bar{x})|\} \leqslant p|\bar{x}|$. Then $t(\bar{x}) = s(\bar{x}) \leftrightarrow B_=(t(\bar{x}), s(\bar{x}); p|\bar{x}|)$ and $t(\bar{x}) < s(\bar{x}) \leftrightarrow B_<(t(\bar{x}), s(\bar{x}); p|\bar{x}|)$ by (9). In the RHS's replace $Bit(i, t(\bar{x})) = 1$ by $C_t^*(\bar{x}; i)$, and quantifiers $Qi < u$ by $Qi < p|\bar{x}|$. $\square$

Lemma 5.2. *For any term* $t(\bar{x})$, $t(\bar{x}) \in \mathscr{C}_{\mathrm{BA}}$.

Proof by induction on the complexity of the term t. Put $t \equiv f(\bar{t})$ with $\bar{t} = t_0, \ldots, t_{n-1}$, and suppose $\bar{t} \subseteq \mathscr{C}_{\mathrm{BA}}$ as IH (induction hypothesis). Let $p|\bar{x}|$ be a polynomial such that $\sum\{|t_k(\bar{x})|: k < n\} \leqslant p|\bar{x}|$. Then by (9) $Bit(i, t) = 1 \leftrightarrow B_f(\bar{t}; p|\bar{x}|; i)$. In the RHS replace $Bit(j, t_k(\bar{x})) = 1$ by $C_{t_k}^*(\bar{x}; j)$, and quantifiers $Qi < u$ by $Qi < p|\bar{x}|$.

Lemma 5.3. *For any* Σ_0^{b}*-formula* $C(\bar{x}) \in L_{\mathrm{BA}}$, $C(\bar{x}) \in \mathscr{C}_{\mathrm{BA}}$.

Proof. By induction on the length of Σ_0^{b}-formula $C(\bar{x}) \in L_{\mathrm{BA}}$. From Lemmata 5.1 and 5.2 we see that the case C is atomic. Consider the case $C(\bar{x}) \equiv \forall y \leqslant |t| C_0(\bar{x}, y)$. Then $C(\bar{x}) \leftrightarrow \forall y \leqslant |t| C_0^*(\bar{x}, y) \leftrightarrow \forall |y| \leqslant \ell \|\bar{x}\| [y \leqslant |t| \rightarrow C_0^*(\bar{x}, y)]$ for a linear form $\ell\|\bar{x}\| \geqslant \|t\|$. By the atomic cases, the part $y \leqslant |t|$ can be stratified. We are done. $\square$

Definition 5.3. For a term $t(\bar{x})$, $C_t^*(\bar{x}; i)$ denotes a stratified (Σ_0^{b}-)formula in $\mathscr{B}_{\mathrm{BA}}(\bar{x}; i)$ so that

$$AID \vdash \forall i < |t(\bar{x})| [Bit(i, t(\bar{x})) = 1 \leftrightarrow C_t^*(\bar{x}; i)]. \tag{10}$$

For a Σ_0^{b}-formula $C(\bar{x}) \in L_{\mathrm{BA}}$, $C^*(\bar{x})$ denotes a stratified (Σ_0^{b}-)formula in $\mathscr{B}_{\mathrm{BA}}(\bar{x};)$ so that

$$AID \vdash C(\bar{x}) \leftrightarrow C^*(\bar{x}). \tag{11}$$

Remark. Note that by the construction, we see that, in the formulae $C_t^*(\bar{x}; i)$ giving ith digit of a term t and in the stratified form $C^*(\bar{x})$ of an a tomic formula, no quantifier $Q|y| \leqslant \ell\|\bar{x}\|$ occurs for a 'second-order' variable y, i.e., a variable in the first list $\bar{x}$.

Next, we define stratified formulae in L_{AID}.

Definition 5.4 (*Stratified formulae in L_{AID}*). Let $\bar{x}$ and $\bar{\imath}$ be sequences of variables with $\bar{x} \cap \bar{\imath} = \emptyset$. Let $\mathscr{B}(\bar{x}; \bar{\imath})$ denote the set of formulae generated as follows:
1. $\mathscr{B}_{\mathrm{BA}}(\bar{x}; \bar{\imath}) \subseteq \mathscr{B}(\bar{x}; \bar{\imath})$.
2. For the inductively defined predicate $A^{\ell,B,\bar{D},I}$ defined from $\ell, B, \bar{D}, I$, $A^{\ell,B,\bar{D},I}(t_1, \ldots, t_n, s) \in \mathscr{B}(\bar{x}; \bar{\imath})$ iff
   (a) Terms $t_1, \ldots, t_n, s$ are variables $y_1, \ldots, y_n, p$ so that $y_1, \ldots, y_n \subseteq \bar{x}$ & $p \in \bar{\imath}$, and
   (b) $B(x_1, \ldots, x_n, i), \bar{D}(x_1, \ldots, x_n, i) \in \mathscr{B}_{\mathrm{BA}}(\bar{x}; \bar{\imath})$.
3. $\mathscr{B}(\bar{x}; \bar{\imath})$ is closed under propositional connectives.
4. If $B(\bar{x}; \bar{\imath}^\frown j) \in \mathscr{B}(\bar{x}; \bar{\imath}^\frown j)$, then $Qj < p|\bar{x}|B(\bar{x}; \bar{\imath}^\frown j) \in \mathscr{B}(\bar{x}; \bar{\imath})$ for any polynomial $p|\bar{x}|$ and $Q \in \{\forall, \exists\}$.
5. If $B(\bar{x}^\frown y; \bar{\imath}) \in \mathscr{B}(\bar{x}^\frown y; \bar{\imath})$, then $Q|y| \leqslant \ell\|\bar{x}\| B(\bar{x}^\frown y; \bar{\imath}) \in \mathscr{B}(\bar{x}; \bar{\imath})$ for any linear form $\ell\|\bar{x}\|$ and $Q \in \{\forall, \exists\}$.

We say that a Σ^{b}_0-formula $B(\bar{x}; \bar{\imath})$ in L_{AID} *is stratified with respect to* $(\bar{x}; \bar{\imath})$ if $B(\bar{x}; \bar{\imath}) \in \mathscr{B}(\bar{x}; \bar{\imath})$. Also we say that a Σ^{b}_0-formula $B(\bar{x})$ in L_{AID} is *stratified with respect to* $\bar{x}$ if $B(\bar{x}) \in \mathscr{B}(\bar{x};\,)$ with the empty list $\bar{\imath} = \emptyset$. When no confusion is likely to occur, we simply write $\mathscr{B}$ for $\mathscr{B}(\bar{x}; \bar{\imath})$.

Let $C(\bar{x})$ be a Σ^{b}_0-formula in L_{AID} with variables $\bar{x}$. (Every free variable in $C(\bar{x})$ need not be in the list $\bar{x}$.) We say that $C(\bar{x})$ is *bitwise computable with respect to* $\bar{x}$ denoted by $C(\bar{x}) \in \mathscr{C}$ if there exists a stratified formula $C^*(\bar{x}) \in \mathscr{B}(\bar{x};\,)$ such that

$$AID \vdash C(\bar{x}) \leftrightarrow C^*(\bar{x}). \tag{11}$$

Lemma 5.4. *For any Σ^{b}_0-formula $C(\bar{x}) \in L_{AID}$, $C(\bar{x}) \in \mathscr{C}$.*

Proof. By induction on the complexity of Σ^{b}_0-formula $C(\bar{x}) \in L_{AID}$. The case $C(\bar{x}) \in L_{\mathrm{BA}}$ is done in Lemma 5.3. Consider the case when $C(\bar{x})$ is a formula $A(\bar{t}(\bar{x}), s(\bar{x}))$ for an inductively defined predicate A.

Case 1: The term s is a variable $x \in \bar{x}$: Then for a polynomial $q|\bar{x}|$ with $|q|\bar{x}|| \geqslant \ell\|\bar{x}\|$, $A(\bar{t}(\bar{x}), x) \leftrightarrow \exists p < q|\bar{x}|[p = x \wedge A(\bar{t}(\bar{x}), p)]$. Therefore it suffices to show $A(\bar{t}(\bar{x}), p) \leftrightarrow A_0(\bar{x}, p)$ for some $A_0(\bar{x}, p) \in \mathscr{B}(\bar{x}; p)$. Let ℓ_0 be a linear form such that $\ell\|\bar{t}(\bar{x})\| \leqslant \ell_0\|\bar{x}\|$, $B_0(\bar{x}, p), \bar{D}_0(\bar{x}, p) \in \mathscr{B}_{\mathrm{BA}}(\bar{x}; p)$ stratified formulae such that $\bar{D}_0(\bar{x}, p) \leftrightarrow \bar{D}(\bar{t}(\bar{x}), p)$ and $B_0(\bar{x}, p) \leftrightarrow B(\bar{t}(\bar{x}), p)$. Then
(A.1) $0 \neq |p| = \ell_0\|\bar{x}\| \to [A(\bar{t}(\bar{x}), p) \leftrightarrow \ell\|\bar{t}(\bar{x})\| = \ell_0\|\bar{x}\| \,\&\, B_0(\bar{x}, p)]$, and
(A.2) $0 \neq |p| < \ell_0\|\bar{x}\| \to [A(\bar{t}(\bar{x}), p)$ iff either $|p| < \ell\|\bar{t}(\bar{x})\| \,\&\, I(\bar{D}_0(\bar{x}, p), A(\bar{t}(\bar{x}), p0), A(\bar{t}(\bar{x}), p1))]$ or $|p| = \ell\|\bar{t}(\bar{x})\| \,\&\, B_0(\bar{x}, p)$.

Therefore for a boolean I_0, an inductively defined $A_0(\bar{x}, p)$ from $\ell_0, B_0, \bar{D}_0, I_0$ is a desired one.

Case 2: Otherwise we have $A(\bar{t}(\bar{x}), s(\bar{x})) \leftrightarrow \exists|p| \leqslant \ell\|\bar{t}(\bar{x})\|(A(\bar{t}(\bar{x}), p) \wedge p = s(\bar{x}))$, and hence the case is reduced to the *Case 1* by Lemma 5.3. $\square$

Definition 5.5. For a Σ^{b}_0-formula $C(\bar{x})$ $C^*(\bar{x})$ denotes a stratified Σ^{b}_0-formula in $\mathscr{B}(\bar{x};\,)$ so that

$$AID \vdash C(\bar{x}) \leftrightarrow C^*(\bar{x}). \tag{11}$$

In Section 8 we need to substitute a formula in a stratified formula. First such a substitution is defined and some elementary facts on it are established.

Definition 5.6 (*Substituting a formula in a stratified formula*). Let $C(y) \in \mathscr{B}(\bar{y};\ \bar{\jmath})$ with $y \in \bar{y}$ and $A_0(\bar{x}, i)$ a Σ^{b}_0-formula and $p|\bar{x}|$ a polynomial. We define a Σ^{b}_0-formula $C(\{i < p|\bar{x}| : A_0(\bar{x}, i)\})$ as follows: Let $lh(p, A_0)$ denote a Σ^{b}_0-definable function in *AID* such that

$$lh(p, A_0) = \begin{cases} \max\{i < p|\bar{x}| : A_0(\bar{x}, i)\} + 1 & \text{if } \exists i < p|\bar{x}| A_0(\bar{x}, i), \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Then $C(\{i < p|\bar{x}|: A_0(\bar{x}, i)\})$ is obtained by replacing $Bit(j, y) = 1$ by $j < p|\bar{x}| \wedge A_0(\bar{x}, j)$ and by replacing $|y|$ by $lh(p, A_0)$.

The substitution $y \mapsto \{i < p|\bar{x}|: A_0(\bar{x}, i)\}$ commutes with propositional connectives.

If $C(y) \equiv Qz < q|y| C_0(z, y)$, then

$$C(\{i < p|\bar{x}|: A_0(\bar{x}, i)\}) \Leftrightarrow_{\mathrm{df}} Qz < q(lh(p, A)) C_0(z, \{i < p|\bar{x}|: A_0(\bar{x}, i)\}),$$

$$Bit(j, \{i < p|\bar{x}|: A_0(\bar{x}, i)\}) = 1 \Leftrightarrow_{\mathrm{df}} j < p|\bar{x}| \wedge A_0(\bar{x}, j).$$

The case $C(y) \equiv A^{\ell, B, \bar{D}, I}(t_1, \ldots, t_{i-1}, y, t_{i+1}, \ldots, t_n, j)$: For simplicity we assume that none of variables $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ is the variable y. Let ℓ' denote a linear form such that if $\ell\|\bar{z}\| = \sum_k c_k\|z_k\| + d$, then

$$\sum_{k \neq i} c_k\|z_k\| + d + c_i|lh(p, A_0)| < \ell'\|z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n, \bar{x}\|.$$

Put

$$B'(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n, \bar{x}, p) \Leftrightarrow_{\mathrm{df}}$$

$$B(z_1, \ldots, z_{i-1}, \{i < p|\bar{x}|: A_0(\bar{x}, i)\}, z_{i+1}, \ldots, z_n, \bar{x}, p),$$

$$\bar{D}'(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n, \bar{x}, p) \Leftrightarrow_{\mathrm{df}}$$

$$\bar{D}(z_1, \ldots, z_{i-1}, \{i < p|\bar{x}|: A_0(\bar{x}, i)\}, z_{i+1}, \ldots, z_n, \bar{x}, p),$$

$B', \bar{D}'$ are Σ^{b}_0-formuale in L_{AID} if $A_0(\bar{x}, i) \notin L_{\mathrm{BA}}$.

Let $A'(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n, \bar{x}, p)$ denote the inductively defined predicate such that for $0 \neq |p| \leqslant \ell'\|z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n, \bar{x}\|$
$A'(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n, \bar{x}, p)$ iff
$|p| = \sum_{k \neq i} c_k\|z_k\| + d + c_i|lh(p, A_0)|\ \&\ B'(\bar{z}, \bar{x}, p)$, or
$|p| < \sum_{k \neq i} c_k\|z_k\| + d + c_i|lh(p, A_0)|\ \&\ I(\bar{D}'(\bar{z}, \bar{x}, p), A'(\bar{z}, \bar{x}, p0), A'(\bar{z}, \bar{x}, p1))$
with $\bar{z} = z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n$.

A' is Σ^{b}_0-definable in *AID* by Lemma 2.4, (Iterated inductive definitions) when $A_0 \notin L_{\mathrm{BA}}$. Then

$$A^{\ell, B, \bar{D}, I}(t_1, \ldots, t_{i-1}, \{i < p|\bar{x}|: A_0(\bar{x}, i)\}, t_{i+1}, \ldots, t_n, j) \Leftrightarrow_{\mathrm{df}}$$

$$A'(t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n, \bar{x}, j).$$

Definition 5.7. For Σ^b_0-formulae $C(y), A_0(\bar{x}, i)$ and a polynomial $p|\bar{x}|$, $C(\{i < p|\bar{x}|: A_0(\bar{x}, i)\})$ denote the formula $C^*(\{i < p|\bar{x}|: A_0(\bar{x}, i)\})$.

Lemma 5.5. *For a stratified formula $C^*(y)$,*

$$AID \vdash y = \{i < p|\bar{x}|: A(\bar{x}, i)\} \rightarrow [C^*(y) \leftrightarrow C^*(\{i < p|\bar{x}|: A(\bar{x}, i)\})],$$

where $y = \{i < p|\bar{x}|: A(\bar{x}, i)\} \quad \Leftrightarrow_{\mathrm{df}} \quad |y| < p|\bar{x}| \,\&\, \forall i < p|\bar{x}|(i \in y \leftrightarrow A(\bar{x}, i))$.

Proof. By induction on the complexity of C^*. □

Lemma 5.6. *Let $C(y)$ be a Σ^b_0-formula in L_{AID}. Let $f(\bar{x})$ be a Σ^b_0-bitdefinable function in AID, and hence Σ^b_0-definable in $AID + \Sigma^b_0-\mathrm{CA}$:*

$$f(\bar{x}) = y \quad \Leftrightarrow_{\mathrm{df}} \quad |y| \leqslant p|\bar{x}| \;\; \& \;\; \forall i < p|\bar{x}|(Bit(i, y) = 1 \leftrightarrow A(\bar{x}, i))(A \in \Sigma^b_0).$$

Then $C(f(\bar{x}))$ is Σ^b_0-definable in AID.

Proof. By induction on the complexity of the stratified form C^* of C using (12). □

Lemma 5.7. 1. *For a term $t(\bar{x})$ let $t^*(\bar{x})$ denote $\{i < |t(\bar{x})|: C^*_t(\bar{x}; i)\}$. For terms $s(z, y)$ and $t(y)$ let $u(y) =_{\mathrm{df}} s(t(y), y) =_{\mathrm{df}} s(z, y)[t(y)/z]$ denote the result of substituting $t(y)$ for z in $s(z, y)$. Then*

$$AID \vdash u^*(y) = s^*(z, y)[t^*(y)/z],$$

where $=$ means that these are coextensional.

2. *For a Σ^b_0-formula $C(z, y)$ and a term $t(y)$,*

$$AID \vdash C(t(y), y) \leftrightarrow C^*(z, y)[t^*(y)/z].$$

Proof. 5.7.1. By Lemma 5.5 and (10) we have $C^*_s(t^*(y); i) \leftrightarrow C^*_s(t; i) \leftrightarrow Bit(i, s(t(y), y)) = 1 \leftrightarrow C^*_u(y; i)$.

5.7.2. This follows from Lemma 5.5 and (11). □

Lemma 5.8. *Let $B(i)$ be a Σ^b_0-formula in which a variable y does not occur, $C(z, y)$ a Σ^b_0-formula, $p|\bar{t}_0|$ a polynomial for some terms $\bar{t}_0$, $t(y), s(y)$ terms and ℓ a linear form. Let $D(y)$ denote the following Σ^b_0-formula:*

$$D(y) \equiv |t(y)| \leqslant \ell \;\|s(y)\| \wedge C(t(y), y) \rightarrow \exists |z| \leqslant \ell \;\|s(y)\| \; C(z, y).$$

Then $AID \vdash D^*(y)[\{i < p|\bar{t}_0|: B(i)\}/y]$.

Proof. By Lemma 5.7 we have $C(t(y), y) \rightarrow C^*(z, y)[t^*(y)/z]$. Let $C_1(y)$ be a stratified formula such that $C_1(y) \leftrightarrow C(t(y), y)$.

Then $C_1(B) \rightarrow C^*(z, y)[t^*(B)/z, B/y]$ for $B = \{i < p|\bar{t}_0|: B(i)\}$ and $t^*(B) = t^*(y)[B/y]$, a Σ^b_0-formula. By $|t^*(B)| \leqslant \ell \|s^*(B)\|$ and Σ^b_0-LCA, Lemma 1.1 we have $\exists |z| \leqslant \ell \|s^*(B)\|$ $\forall i < |t^*(B)| \; [i \in z \leftrightarrow t^*(B)]$. □

6. Frege system simulates *AID*

In this section we show that any Σ^b_0-theorem in *AID* yields true boolean sentences of which $\mathscr{F}$ has polysize proofs.

For each stratified Σ^b_0-formula $B(\bar{x};\bar{\imath}) \in \mathscr{B}(\bar{x};\bar{\imath})\,(\bar{x}=x_1,\dots,x_n)$ in L_{AID}, we define a valuation $\sigma_{\bar{x}}$: $\{p^k_j : 1\leqslant k\leqslant n,\ j<|x_k|\} \to \{\top,\bot\}$ and a boolean formula $\langle B(\bar{x};\bar{\imath})\rangle$ so that

1. Atoms occurring in $\langle B(\bar{x};\bar{\imath})\rangle$ are among the atoms $\bar{p}^1,\dots,\bar{p}^n$ where $\bar{p}^k = p^k_0,\dots, p^k_{m_k-1}$ with $m_k = |x_k|$.
2. The bitgraph of the function $\varphi_B : (\bar{x};\bar{\imath}) \mapsto \langle B(\bar{x};\bar{\imath})\rangle$ is Σ^b_0-definable in *AID*, i.e., there exists a Σ^b_0-formula $A_B(\bar{x};\bar{\imath},j)$ in L_{AID} for each B such that
$$A_B(\bar{x};\bar{\imath},j) \leftrightarrow Bit(j,\langle B(\bar{x};\bar{\imath})\rangle)=1,$$
where the boolean formula $\langle B(\bar{x};\bar{\imath})\rangle$ is coded by 0-1 words as in [7].
3. $$AID \vdash B(\bar{x};\bar{\imath}) \leftrightarrow TRUE(\langle B(\bar{x};\bar{\imath})\rangle;\sigma_{\bar{x}}),$$
where RHS means that the boolean formula $\langle B(\bar{x};\bar{\imath})\rangle$ is true under the valuation $\sigma_{\bar{x}}$ defined by
$$\sigma_{\bar{x}}(p^k_j) = \begin{cases} \top & \text{if } Bit(j,x_k)=1, \\ \bot & \text{if } Bit(j,x_k)=0. \end{cases}$$
Alternatively, we can define $\langle B(\bar{x};\bar{\imath})\rangle$ as a sentence which is the result of replacing each atom p^k_j by $\sigma_{\bar{x}}(p^k_j)$.

Let $\varphi(\bar{p}^y)$ be a boolean formula with atoms $\bar{p}^y = p^y_0,\dots,p^y_{|y|-1}$ and $\sigma_y : \bar{p}^y \to \{\top,\bot\}$ the valuation defined by
$$\sigma_y(p^y_j) = \begin{cases} \top & \text{if } Bit(j,y)=1, \\ \bot & \text{if } Bit(j,y)=0. \end{cases}$$
Then $(\varphi(\bar{p}^y);\sigma_y)$ denotes the result of replacing each atom p^y_j by $\sigma_y(p^y_j) \in \{\top,\bot\}$.

Definition 6.1 (*Translation into boolean formulae*)**.** In the following we define inductively a boolean formula $\langle B(\bar{x};\bar{\imath})\rangle$ for $B(\bar{x};\bar{\imath}) \in \mathscr{B}(\bar{x};\bar{\imath})$. First $\langle B(\bar{x};\bar{\imath})\rangle$ is defined for a formula $B(\bar{x};\bar{\imath}) \in L_{\mathrm{BA}}$ in which no quantifier $|y|\leqslant\ell\|\bar{x}\|$ occurs. This includes the stratified forms of atomic formulae in L_{BA}, e.g., of the formula $|y|\leqslant\ell\|\bar{x}\|$, cf. **Remark** after Definition 5.3. Then a boolean formula is defined for any formula in L_{BA}, and finally for any formula in L_{AID}.

1. $$\langle Bit(i,x_k)=1\rangle = \begin{cases} p^k_i & \text{if } i<|x_k|, \\ \bot & \text{otherwise.} \end{cases}$$
$\langle Bit(i,j)=1\rangle, \langle p(\bar{\imath})=q(\bar{\imath})\rangle, \langle p(\bar{\imath})<q(\bar{\imath})\rangle$ are defined to be $\top$ or $\bot$ if the formula is true or false, resp.

2. Inductively, defined predicate $A^{\ell,B,\bar{D},I}$.
   (A.0) $\neg[0 \neq |p| \leqslant \ell\|\bar{x}\|]$: $\langle A^{\ell,B,\bar{D},I}(\bar{x}, p)\rangle =_{\mathrm{df}} \bot$.
   (A.1) $0 \neq |p| \leqslant \ell\|\bar{x}\|$ & $|p| = \ell\|\bar{x}\|$: $\langle A^{\ell,B,\bar{D},I}(\bar{x}, p)\rangle =_{\mathrm{df}} \langle B(\bar{x}, p)\rangle$.
   (A.2) $0 \neq |p| \leqslant \ell\|\bar{x}\|$ & $|p| < \ell\|\bar{x}\|$:
$$\langle A^{B,\bar{D},I,\ell}(\bar{x}, p)\rangle =_{\mathrm{df}} \langle I(\bar{D}(\bar{x}, p), A^{\ell,B,\bar{D},I}(\bar{x}, p0), A^{\ell,B,\bar{D},I}(\bar{x}, p1))\rangle.$$
3. $\langle\cdot\rangle$ commutes with propositional connectives.
4. $\langle \forall j < p|\bar{x}| B(\bar{x}; \bar{\imath}^\frown j)\rangle = \bigwedge\{\langle B(\bar{x}; \bar{\imath}^\frown j)\rangle: j < p|\bar{x}|\}$ and
   $\langle \exists j < p|\bar{x}| B(\bar{x}; \bar{\imath}^\frown j)\rangle = \bigvee\{\langle B(\bar{x}; \bar{\imath}^\frown j)\rangle: j < p|\bar{x}|\}$.
5. $\langle \forall |y| \leqslant \ell\|\bar{x}\| B(\bar{x}^\frown y; \bar{\imath})\rangle = \bigwedge\{(\langle |y| \leqslant \ell\|\bar{x}\|\rangle \to \langle B(\bar{x}^\frown y; \bar{\imath})\rangle; \sigma_y): |y| \leqslant \ell\|\bar{x}\|\}$ and
   $\langle \exists |y| \leqslant \ell\|\bar{x}\| B(\bar{x}^\frown y; \bar{\imath})\rangle = \bigvee\{(\langle |y| \leqslant \ell\|\bar{x}\|\rangle \wedge \langle B(\bar{x}^\frown y; \bar{\imath})\rangle; \sigma_y): |y| \leqslant \ell\|\bar{x}\|\}$.

It is straightforward to see that the size of the boolean formula $\langle B(\bar{x}; \bar{\imath})\rangle$ is bounded by a polynomial in $\bar{x}$ for each B. The Σ_0^{b}-bitdefinability of $\langle B(\bar{x}; \bar{\imath})\rangle$ follows from a tree induction in *AID*.

Theorem 6.1. *For any Σ_0^{b}-formula $B(\bar{x})$, if $AID \vdash B(\bar{x})$, then there exists a polynomial $p|\bar{x}|$ and a Σ_0^{b}-formula $P(\bar{x}, i)$ such that*

$AID \vdash \{i < p|\bar{x}|: P(\bar{x}, i)\}$ is a Frege proof of $\langle B^*(\bar{x})\rangle$

and hence by Σ_0^{b}-CA, $\exists |y| \leqslant p|\bar{x}|\ \forall i < p|\bar{x}|\ (i \in y \leftrightarrow P(\bar{x}, i))$, $AID + \Sigma_0^{\mathrm{b}}\text{-CA} \vdash \mathscr{F} \vdash^{p|\bar{x}|} \langle B^*(\bar{x})\rangle$ for an equivalent stratified formula B^*.

Note that Σ_0^{b}-CA is needed here only because of a hidden existential bounded quatifier in $\mathscr{F} \vdash^{p|\bar{x}|}$.

Theorem 6.1 is seen as in [18, pp. 148–154] by showing first, in *AID*, that there exists a polysize Frege proof of $\langle [y = t \to (B(y) \leftrightarrow B(t))]^*\rangle$ for each Σ_0^{b}-formula $B(y)$ and each term t with a fresh variable y. The latter is shown by induction on the formula $B(y)$ and the term t.

7. Bounded arithmetics for Frege system

In this section we introduce some systems of bounded arithmetic in the language L_{BA}, i.e., without inductively defined predicates $A^{\ell,B,\bar{D},I}$ which are equivalent to *AID*. These systems contain a base fragment Σ_0^{b}-LIND of bounded arithmetic in the language L_{BA}.

Definition 7.1. Σ_0^{b}-RD (Σ_0^{b}-Recursive definitions) denotes the axiom schema whose instances are of the following form: for $\Sigma_0^{\mathrm{b}}\, B, \bar{D}$, a boolean I and a linear form ℓ, cf. (A.1) and (A.2) in Definition 1.2,

$$\begin{aligned}&\forall \bar{x}\ \exists |y| \leqslant 2^{\ell\|\bar{x}\|} \forall |i| \leqslant \ell\|\bar{x}\| [\{0 \neq |i| = \ell\|\bar{x}\| \to (i \in y \leftrightarrow B(\bar{x}, i))\}\\ &\quad \wedge \{0 \neq |i| < \ell\|\bar{x}\| \to (i \in y \leftrightarrow I(\bar{D}(\bar{x}, i), i0 \in y, i1 \in y))\}]\end{aligned} \tag{13}$$

where $i \in y \leftrightarrow Bit(i, y) = 1$.

Lemma 7.1. Σ_0^{b}--RD $\vdash \Sigma_0^{\mathrm{b}}$-CA.

Proof. Let B be a Σ_0^{b}-formula and $p|x|$ a polynomial. Let ℓ denote a linear form such that $|p|x||<\ell\|x\|$. Pick a y by using Σ_0^{b}-RD so that $|i|=\ell\|x\| \rightarrow (i\in y \leftrightarrow B(i_0)\wedge i_0<p|x|)$ for $i_0=i[0,\ell\|x\|-1)$. Then $z=y[2^{\ell\|x\|-1},|y|)$ is a required set since $i\in z \leftrightarrow 2^{\ell\|x\|-1}+i\in y \leftrightarrow B(i)$ for $i<p|x|$. $\square$

Lemma 7.2. *For each inductively defined predicate $A=A^{\ell,B,\bar{D},I}$ in L_{AID}, there exists a Δ_1^{b}-formula A' in Σ_0^{b}-RD, i.e., there are a Σ_1^{b} A_Σ and a Π_1^{b} A_Π such that Σ_0^{b}-RD $\vdash A' \leftrightarrow_{\mathrm{df}} A_\Sigma \leftrightarrow A_\Pi$, so that for any formula $\varphi(A,\ldots)$ in L_{AID},*

$$AID\vdash \varphi(A,\ldots) \Rightarrow \Sigma_0^{\mathrm{b}}\text{-RD}+\Delta_1^{\mathrm{b}}\text{-LIND}\vdash \varphi(A',\ldots),$$

where each A is replaced by the corresponding Δ_1^{b} A'.

Note that Σ_0^{b}-RD+Δ_1^{b}-LIND $\subseteq \Sigma_0^{\mathrm{b}}$-RD+$\Delta_1^{\mathrm{b}}$-CA $\subseteq \Sigma_0^{\mathrm{b}}$-RD+$\Sigma_1^{\mathrm{b}}$-AC. *Thus* AID+Δ_1^{b}-CA *and* $AID+\Sigma_1^{\mathrm{b}}$-AC *are interpretable in* Σ_0^{b}-RD $+\ \Delta_1^{\mathrm{b}}$-CA *and* Σ_0^{b}-RD $+\ \Sigma_1^{\mathrm{b}}$-AC, *resp.*

Proof. We show $A=A^{\ell,B,\bar{D},I}$ is Δ_1^{b}-definable in Σ_0^{b}-RD. Then $\Sigma_0^{\mathrm{b}}(L_{AID})$-LIND in L_{AID} turns into $\Delta_1^{\mathrm{b}}(L_{BA})$-LIND.

Let $p|\bar{x}|$ be a polynomial such that $2^{\ell\|\bar{x}\|}\leqslant p|\bar{x}|$. Let $Demo(y,\bar{x})$ denote the following Σ_0^{b}-formula in L_{BA}, cf. (13):

$$\begin{aligned}&|y|\leqslant p|\bar{x}| \ \& \ \forall|i|\leqslant\ell\|\bar{x}\|[\{0\neq|i|=\ell\|\bar{x}\|\rightarrow(i\in y\leftrightarrow B(\bar{x},i))\}\\ &\quad\wedge\{0\neq|i|<\ell\|\bar{x}\|\rightarrow(i\in y\leftrightarrow I(\bar{D}(\bar{x},i),i0\in y,i1\in y))\}].\end{aligned}$$

By Σ_0^{b}-RD we have $\forall\bar{x}\exists\,|y|\leqslant p|\bar{x}|\,Dem(y,\bar{x})$. From Σ_0^{b}-LIND we see that such a demonstration tree y is unique:

$$Demo(y,\bar{x}) \ \& \ Demo(z,\bar{x})\,\&\,0\neq|i|\leqslant\ell\|\bar{x}\|\rightarrow(i\in y\leftrightarrow i\in z).$$

Therefore in Σ_0^{b}-RD

$$\begin{aligned}&A_\Sigma(\bar{x},i)\leftrightarrow_{\mathrm{df}}\exists|y|\leqslant p|\bar{x}|[Demo(y,\bar{x})\,\&\,i\in y]\,\&\,0\neq|i|\leqslant\ell\|\bar{x}\|\\ &\leftrightarrow\\ &A_\Pi(\bar{x},i)\leftrightarrow_{\mathrm{df}}\forall|y|\leqslant p|\bar{x}|[Demo(y,\bar{x})\rightarrow i\in y]\,\&\,0\neq|i|\leqslant\ell\|\bar{x}\|. \qquad \square\end{aligned}$$

8. Realizations of Σ_1^{b}-consequences

In this section we show that Σ_1^{b}-consequences in $AID+\Sigma_0^{\mathrm{b}}$-CA or Σ_0^{b}-RD $+\ \Sigma_1^{\mathrm{b}}$-AC can be realized by a Σ_0^{b}-set $\{i<p|\bar{x}|\colon A(\bar{x},i)\}$.

Recall that C^* denotes a stratified formula which is equivalent to a given Σ_0^{b}-formula C, cf. Definition 5.5.

Lemma 8.1. *Let $C(z,\bar{x})$ be a Σ^b_0-formula and $p_0|\bar{x}|$ a polynomial. If AID* + Σ^b_0-CA $\vdash \exists|z|\leqslant p_0|\bar{x}|C(z,\bar{x})$, *then there exists a Σ^b_0-formula $A(\bar{x},i)$ such that AID* $\vdash C^*(\{i<p_0|\bar{x}|\colon A(\bar{x},i)\},\bar{x})$. *In particular AID* + Σ^b_0-CA *is Σ^b_0-conservative over AID.*

Proof. This is an analogue to the fact about the subsystem ACA_0 of second-order arithmetic vs. PA. Hence the idea of a proof is to replace a 'set' variable y by its Σ^b_0-definition $\{i<p|\bar{x}|\colon B(i)\}$ when an instance of Σ^b_0-CA, $\exists|y|\leqslant p|\bar{x}|\forall i<p|\bar{x}|(i\in y\leftrightarrow B(i))$ occurs.

Formulate *AID* + Σ^b_0-CA in Gentzen's sequent calculus. Sequents are denoted $\Gamma\Rightarrow\Delta$. Σ^b_0-LIND and Σ^b_0-CA are replaced by the following inference rules for $B\in\Sigma^b_0$ and eigenvariables y:

$$\frac{B(y),\Gamma\Rightarrow\Delta,B(y+1)}{B(0),\Gamma\Rightarrow\Delta,B(|t|)},$$

$$\frac{|y|\leqslant p|\bar{x}|,\forall i<p|\bar{x}|(i\in y\leftrightarrow B(i)),\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow\Delta}.$$

First eliminate cuts partially to get a proof in which every sequent is in $\Sigma^b_0\cup s\Sigma^b_1$.

We have $C(A_0)\vee C(A_1)\to C(A)$, where $A_j=\{i<p|\bar{x}|:A_j(\bar{x},i)\}$ and $A(\bar{x},i)\Leftrightarrow_{\mathrm{df}}(C(A_0)\wedge A_0(\bar{x},i))\vee(\neg C(A_0)\wedge A_1(\bar{x},i))$.

This cares *Contraction* : *right*. $\exists\leqslant$: *right* is seen from Lemma 5.7.2.

Assume

$$|y|\leqslant p_0|t_0(\bar{x})|,\forall i<p_0|t_0|(i\in y\leftrightarrow B(i)),\Gamma\Rightarrow\Delta,C(V,\bar{x})$$

with $V=\{i<p|\bar{x},y|\colon A(\bar{x},y,i)\}$. Replace each formula $C_0(y)$ in a proof of this sequent by $C_0^*(\{i<p_0|t_0(\bar{x})|\colon B(i)\})$. Use Lemma 5.8 to handle $\exists\leqslant$: *right* and $\forall\leqslant$: *left*. Therefore, *AID* proves $\Gamma\Rightarrow\Delta,C(\{i<p'|\bar{x}|\colon A'(\bar{x},i)\},\bar{x})$ with a polynomial p' and $A'(\bar{x},i)\Leftrightarrow_{\mathrm{df}}A^*(\bar{x},\{i<p_0|t_0(\bar{x})|\colon B(i)\},i)$. $\square$

Lemma 8.2. *For a Σ^b_0-formula $B(y,\bar{x})$ and a polynomial $p|\bar{x}|$, if* Σ^b_0-RD + Σ^b_1-AC $\vdash \forall\bar{x}\exists|y|\leqslant p|\bar{x}|B(y,\bar{x})$, *then there exists a Σ^b_0-formula $A(\bar{x},i)$ in L_{AID} such that AID* $\vdash B^*(\{i<p|\bar{x}|\colon A(\bar{x},i)\},\bar{x})$.

Proof. Formulate the system Σ^b_0-RD + Σ^b_1-AC = Σ^b_0-RD + Σ^b_0-AC in Gentzen's sequent calculus:

1. Initial sequents: logical ones $A\Rightarrow A$ ($A\in\Sigma^b_0$), *BASIC*, Bit Extensionality Axiom, Σ^b_0-LIND, Σ^b_0-RD. These are in $s\Sigma^b_1$.
2. Inference rules *LKB* and Σ^b_0-AC: for $B\in\Sigma^b_0$

$$\frac{i<p|t|,\Gamma\Rightarrow\Delta,\exists|y|\leqslant q|t|B(i,y)}{\Gamma\Rightarrow\Delta,\exists|z|\leqslant p|t|\cdot q|t|\forall i<p|t|B(i,z_i)}.$$

Suppose Σ^b_0-RD + Σ^b_1-AC $\vdash\forall\bar{x}\exists|y|\leqslant p|\bar{x}|B(y,\bar{x})$. Eliminate cuts partially. There is a proof of $\Rightarrow\exists|y|\leqslant p|\bar{x}|B(y,\bar{x})$ such that every formula in it is either Σ^b_0 or $s\Sigma^b_1$. That is to say, every sequent in it is of the form

$$\{\exists |z_i| \leqslant q_i|\bar{t}_i| C_i(z_i,\bar{x}) \colon i<n\}, \Pi \Rightarrow \Lambda, \{\exists |y_j| \leqslant p_j|\bar{s}_j| B_j(y_j,\bar{x}) \colon j<m\}$$

with $\{C_i\}_{i<n}, \Pi, \Lambda, \{B_j\}_{j<m} \subseteq \Sigma^{\mathrm{b}}_0$.

We show, by induction on the depth of proofs, that there exist Σ^{b}_0-formulae $A_j(\bar{x},\bar{z},i)\,(j<m, \bar{z}=z_0,\dots,z_{n-1})$ such that for $V_j = \{i<p_j|\bar{s}_j| \colon A_j(\bar{x},\bar{z},i)\}$

$$\{|z_i| \leqslant q_i|\bar{t}_i| \wedge C_i(z_i,\bar{x}) \colon i<n\}, \Pi \Rightarrow \Lambda, \{B_j(V_j,\bar{x}) \colon j<m\}$$

is provable in *AID*.

Case 1: Σ^{b}_0-RD:

$$\begin{aligned} &\Rightarrow \exists |y| \leqslant 2^{\ell \| \bar{t} \|}\ \forall |i| \leqslant \ell \| \bar{t} \| [\{0 \neq |i| = \ell \| \bar{t} \| \to (i \in y \leftrightarrow B(\bar{t},i))\} \\ &\quad \wedge \{0 \neq |i| < \ell \| \bar{t} \| \to (i \in y \leftrightarrow I(\bar{D}(\bar{t},i), i0 \in y, i1 \in y))\}] \end{aligned}$$

for a sequence $t=\bar{t}(\bar{x})$ of terms. Then put $A(\bar{x},i) \leftrightarrow_{\mathrm{df}} A^{\ell,B,\bar{D},I}(\bar{t}(\bar{x}),i)$ for the inductively defined $A^{\ell,B,\bar{D},I}$.

Case 2: Contraction: As in the proof of Lemma 8.1.

Case 3: Σ^{b}_0-AC: By IH we have $i<p|t|,\ \Gamma \Rightarrow \Delta,\ B(i,\{j<q|t| \colon A(\bar{x},i,j)\})$, where we assume $\Gamma, \Delta \in \Sigma^{\mathrm{b}}_0$ for simplicity. Then $\Gamma \Rightarrow \Delta,\ \forall i<p|t| B(i,Z_i)$ for

$$Z = \{k < p|t| \cdot q|t| \colon \exists i<p|t| \exists j<q|t| (k = i \cdot q|t| + j \wedge A(\bar{x},i,j))\}$$

and $Z_i = Z[i \cdot q|t|, (i+1)\cdot q|t|)$.

Case 4: Cut: Infer

$$\exists |z| \leqslant q|t| C(z,\bar{x}), \Gamma, \Pi \Rightarrow \Delta, \Lambda, \exists |y| \leqslant p|s| B(y,\bar{x})$$

from

$$\exists |z| \leqslant q|t| C(z,\bar{x}), \Gamma \Rightarrow \Delta, \exists |u| \leqslant r|t'| D(u,\bar{x})$$

and

$$\exists |u| \leqslant r|t'| D(u,\bar{x}), \Pi \Rightarrow \Lambda, \exists |y| \leqslant p|s| B(y,\bar{x}).$$

By IH we have

$$|z| \leqslant q|t|, C(z,\bar{x}), \Gamma \Rightarrow \Delta, D(\{i<r|t'| \colon A_0(\bar{x},z,i)\},\bar{x})$$

and

$$|u| \leqslant r|t'|, D(u,\bar{x}), \Pi \Rightarrow \Lambda, B(\{i<p|s| \colon A_1(\bar{x},u,i)\},\bar{x}).$$

In a proof of the latter sequent, substituting $\{i<r|t'| \colon A_0(\bar{x},z,i)\}$ for the variable u we get

$$D(\{i<r|t'| \colon A_0(\bar{x},z,i)\},\bar{x}), \Pi \Rightarrow \Lambda, B(\{i<p|s| \colon A_1(\bar{x},u,i)\},\bar{x})$$

for $A(\bar{x},z,i) \Leftrightarrow_{\mathrm{df}} A_1(\bar{x},\{i<r|t'|: A_0(\bar{x},z,i)\},i)$. By a cut with the cut formula $D(\{i<r|t'|: A_0(\bar{x},z,i)\},\bar{x})$ we get

$$|z| \leqslant q|t|, C(z,\bar{x}), \Gamma, \Pi \Rightarrow \Delta, \Lambda, B(\{i<p|s|: A(\bar{x},z,i)\},\bar{x}). \qquad \square$$

Theorem 8.1. 1. Σ^b_0-RD + Σ^b_1-AC *is* Σ^b_1-*conservative over* $AID + \Sigma^b_0$-CA.

2. Σ^b_0-RD + Σ^b_1-AC *is* Σ^b_0-*conservative over* AID.

3. *Every* $s\Delta^b_1$-*formula in* Σ^b_0-RD + Σ^b_1-AC *is* Σ^b_0-*definable in* AID: *for strict* Σ^b_1-*formulae* $A,B \in s\Sigma^b_1$, *if* Σ^b_0-RD + Σ^b_1-AC $\vdash A \leftrightarrow \neg B$, *then* $AID \vdash A \leftrightarrow A'$ *for a* Σ^b_0 A' *in* L_{AID}.

Proof.
8.1.1. Let $C(\bar{x})$ be a Σ^b_1-formula provable in Σ^b_0-RD + Σ^b_1-AC. Let B be a Σ^b_0-formula so that for a polynomial $p|\bar{x}|$,

$$\Sigma^b_1\text{-AC} \vdash C(\bar{x}) \leftrightarrow \exists|y| \leqslant p|\bar{x}|B(y,\bar{x}).$$

By Lemma 8.2 and Σ^b_0-CA, $\exists|y| \leqslant p|\bar{x}|$ $(y=\{i<p|\bar{x}|: A(\bar{x},i)\})$, we have $AID+\Sigma^b_0$-CA $\vdash \exists|y| \leqslant p|\bar{x}|B(y,\bar{x})$. Since $\exists|y| \leqslant p|\bar{x}|B(y,\bar{x}) \rightarrow C(\bar{x})$ is provable without Σ^b_1-AC, we conclude $AID + \Sigma^b_0$-CA $\vdash C(\bar{x})$.

8.1.3. Suppose Σ^b_0-RD + Σ^b_1-AC $\vdash A(\bar{x}) \leftrightarrow \neg B(\bar{x})$ for $A,B \in s\Sigma^b_1$. By Lemma 8.2 there exists a Σ^b_0-formula $A_0(\bar{x},i)$ such that

$$AID \vdash (0 \in y \leftrightarrow A_0(\bar{x},0)) \rightarrow (y=1 \wedge A(\bar{x})) \vee (y=0 \wedge B(\bar{x}))$$

By Theorem 8.1.2 we have $B(\bar{x}) \rightarrow \neg A(\bar{x})$. Therefore $AID \vdash A_0(\bar{x},0) \leftrightarrow A(\bar{x})$. $\square$

Corollary 8.1. 1. *If a function of polynomial growth rate is* Σ^b_1-*definable in* Σ^b_0-RD + Σ^b_1-AC *in the sense of* [4], *then the function is in* $\mathscr{F}$*ALOGTIME.*

2. *If a predicate is* Δ^b_1-*definable in* Σ^b_0-RD + Σ^b_1-AC, *then the predicate is in ALOGTIME.*

Corollary 8.1.2, Proposition 1.1 and Theorem 3.1 yield:

Theorem 8.2. *For a predicate* A,

$$\begin{aligned} A \in ALOGTIME &\Leftrightarrow A \text{ is } \Sigma^b_0\text{-definable in } AID \\ &\Leftrightarrow A \text{ is } \Delta^b_1\text{-definable in } \Sigma^b_0\text{-RD} + \Sigma^b_1\text{-AC}. \end{aligned}$$

Theorem 8.3. *For a* Σ^b_0-*formula* $B(y,\bar{x})$ *and a polynomial* $p|\bar{x}|$, *if*

$$\Sigma^b_0\text{-RD} + \Sigma^b_1\text{-AC} \vdash \forall\bar{x}\exists\, |y| \leqslant p|\bar{x}|B(y,\bar{x}),$$

then there exist Σ^b_0-*formulae* $A(\bar{x},i), P(\bar{x},i)$ *in* L_{AID} *and a polynomial* $q|\bar{x}|$ *such that*

$$AID \vdash \{i<q|\bar{x}|: P(\bar{x},i)\} \text{ is a Frege proof of } \langle B^*(\{i<p|\bar{x}|: A(\bar{x},i)\},\bar{x})\rangle,$$

and hence $AID + \Sigma^b_0$-CA $\vdash \mathscr{F} \vdash^{q|\bar{x}|} \langle B^*(\{i<p|\bar{x}|: A(\bar{x},i)\},\bar{x})\rangle$.

9. Clote's *ALV* and *AID*

In [10] Clote defines a function algebra N_0 and shows that N_0 is equal to $\mathscr{F}$ *ALOGTIME*, the class of *ALOGTIME*-computable functions. Then he [11] introduces an equational system *ALV* based on N_0, and shows that any provable equation in *ALV* can be transformed into boolean sentences so that $\mathscr{F}$ has polysize proofs of these sentences.

In this section we show that $AID + \Sigma^{\mathrm{b}}_0$-CA is equivalent to *ALV* in the sense that there exist Σ^{b}_1-faithful interpretations between $AID + \Sigma^{\mathrm{b}}_0$-CA and a quantified theory *QALV*.

9.1. $AID + \Delta^{\mathrm{b}}_1$-CA *contains ALV*

In this subsection we show that (the graph of) each function $f \in N_0$ is Δ^{b}_1-definable in $AID + \Delta^{\mathrm{b}}_1$-CA. Hence via Clote's result $N_0 = \mathscr{F}ALOGTIME$ in [10] we get:

Theorem 9.1. *For each* $f \in \mathscr{F}ALOGTIME = N_0$, *there exists a* Δ^{b}_1-*formula* $G_f(\bar{x}, y)$ *in* $AID + \Delta^{\mathrm{b}}_1$-CA *and a polynomial* p_f *so that* $G_f(\bar{x}, y)$ *defines the graph of* f *in the standard model,* $AID + \Delta^{\mathrm{b}}_1\text{-CA} \vdash \forall \bar{x} \exists! y G_f(\bar{x}, y)$ *and* $AID + \Delta^{\mathrm{b}}_1\text{-CA} \vdash G_f(\bar{x}, y) \to |y| \leqslant p_f\,|\bar{x}|$.

Further the definitions are intensionally correct. Namely:

Theorem 9.2. *Using the definition* $G_f(i, \bar{x})$ *of the graph of* $f \in N_0$ *the defining equations of* f *are derivable in* $AID + \Delta^{\mathrm{b}}_1$-CA.

Now we prove Theorems 1 and 2 by induction on the construction of $f \in N_0$.

Initial functions. These are zero $\mathrm{o}(x) = 0$, successor functions $xi = s_i x = 2\cdot x + i$ for $i < 2$, projections $i^n_k(x_1, \ldots, x_n) = x_k$, $Bit(i, x)$, # and the function *tree*. Except the last one *tree*, which is NC^1-complete for AC^0-reduction, cf. [10], the assertions are clear.

The function *tree* takes values $0, 1$ and so we regard it as a predicate. Then the predicate *tree* is defined from the auxiliary functions $and(x)$, $or(x)$ as follows: First set $and(0) = or(0) = 0$ & $and(x) = or(x) = 1$ for $1 \leqslant x \leqslant 3$. For $x > 0$

$$\begin{array}{ll} and(x00) = and(x)0 & or(x00) = or(x)0, \\ and(x10) = and(x)0 & or(x10) = or(x)1, \\ and(x01) = and(x)0 & or(x01) = or(x)1, \\ and(x11) = and(x)1 & or(x11) = or(x)1. \end{array}$$

Then

$$tree(x) \Leftrightarrow \begin{cases} parity(x) =_{\mathrm{df}} Bit(0, x) = 1 & \text{if } x < 16, \\ tree(or(and(x))) & \text{otherwise.} \end{cases}$$

Therefore, for $x > 1$, $tree(x)$ is the predicate obtained by evaluating a perfect *and*/*or* tree on the $4^{\lfloor \log_4(|x| \dot{-} 1) \rfloor}$ many least significant bits of x. We define a predicate

$Tree(x, p)$ inductively so that $tree(x) \leftrightarrow Tree(x, 1)$. Put

$$y = x[0, 4^{\lfloor \log_4(|x| \dot{-} 1) \rfloor})$$

for $x > 1$ and put $y = 0$ for $x \leqslant 1$.

If $x > 1$, then $|y| = 4^{\lfloor \log_4(|x| \dot{-} 1) \rfloor} = \max\{u < |x| : \exists z(u = 4^z)\}$ and hence

$$\|y\| \dot{-} 1 = \begin{cases} \lfloor \frac{\|x\| \dot{-} 2}{2} \rfloor & \text{if } \exists k(|x| = 2^k)[\leftrightarrow \forall i < \|x\| \dot{-} 1(Bit(i, |x|) = 0)], \\ \lfloor \frac{\|x\| \dot{-} 1}{2} \rfloor & \text{otherwise.} \end{cases} \tag{14}$$

Also $\|y\| \dot{-} 1 \leqslant \|x\|$.

(T.0) $Tree(x, p) \rightarrow 0 \neq |p| \leqslant \|x\| + 1$.

(T.1) The case $\|y\| \leqslant |1p| \leqslant \|x\| + 1$: $Tree(x, 1p) \leftrightarrow Bit(p[0, \|y\| \dot{-} 1), x) = 1$

(T.2) The case $0 \neq |p| < \|y\|$: $Tree(x, p)$ iff
either $\{|p|$ is odd & $(Tree(x, p0) \vee Tree(x, p1))\}$
or $\{|p|$ is even & $(Tree(x, p0) \wedge Tree(x, p1))\}$.

Note that by (14) this defines the predicate $Tree = A^{\ell, B, \bar{D}, I}$ in L_{AID} for some Σ^b_0-formulae $B, \bar{D}$ in L_{BA} and a boolean I with $\ell\|x\| = \|x\| + 1$.

Now we show $Tree(x, 1)$ enjoys the defining axioms of *tree* in *AID*. First assume $x < 16$. Then $|x| \leqslant 4$ and hence $|y| = 1$ & $\|y\| \dot{-} 1 = 0$. By (T.1) with $p = 0$ we have $Tree(x, 1) \leftrightarrow Bit(0, x) = parity(x) = 1$.

Next, consider the case $x \geqslant 16$. We have to show

$$Tree(x, 1) \leftrightarrow Tree(or(and(x)), 1).$$

We understand this formula is an abbreviation for the Δ^b_1-formula

$$\begin{aligned} &\forall |y| \leqslant |x|[y = or(and(x)) \rightarrow (Tree(x, 1) \leftrightarrow Tree(y, 1))] \\ &\quad \leftrightarrow \exists |y| \leqslant |x|[y = or(and(x)) \ \& \ (Tree(x, 1) \leftrightarrow Tree(y, 1))] \end{aligned}$$

for a Σ^b_0-formula $y = or(and(x))$.

Observe that $|or(x)| = |and(x)| = \lfloor |x|/2 \rfloor + parity(|x|)$ for $x > 3$. We show the following Claim.

Claim 9.1. *Put*

$$z = 2\lfloor \log_4(|or(and(x))| \dot{-} 1) \rfloor = 2\lfloor \log_4(|x| \dot{-} 1) \rfloor - 2.$$

For $x \geqslant 16$ and any odd $|1p| \leqslant z + 1$

$$Tree(x, 1p) \leftrightarrow Tree(or(and(x)), 1p).$$

Proof of Claim 9.1 by induction on $z + 1 - |1p|$. Here we use Δ^b_1–LIND. This follows from Δ^b_1-CA. If $|1p| < z + 1$, then the Claim follows from IH and (T.2). Suppose $z + 1 = |1p|$. Then by (T.1) we have $Tree(or(and(x)), 1p) \leftrightarrow Bit(p, or(and(x))) = 1$.

On the other hand, we have by (T.2) and (T.1)

$$\begin{aligned}&Tree(x,1p)\\&\quad\leftrightarrow (Tree(x,1p00)\wedge Tree(x,1p01))\vee(Tree(x,1p10)\wedge Tree(x,1p11))\\&\quad\leftrightarrow (Bit(p00,x)=1\wedge Bit(p01,x)=1)\vee(Bit(p10,x)=1\wedge Bit(p11,x)=1).\end{aligned}$$

Put $n=4k+j=|x|\geqslant 5$ for $j<4$. We show

$$[j\neq 0\Rightarrow 4p+3<4k]\ \&\ [j=0\Rightarrow 4p+3<4k-4]. \tag{15}$$

By $|1p|=2\lfloor\log_4(n\dot{-}1)\rfloor-1$ we have $|p|\leqslant 2\lfloor\log_4(n\dot{-}1)\rfloor-2$ and hence $p+1\leqslant 2^{2\lfloor\log_4(n\dot{-}1)\rfloor-2}$. Therefore $4p+4\leqslant 2^{2\lfloor\log_4(n\dot{-}1)\rfloor}=4^{\lfloor\log_4(n\dot{-}1)\rfloor}$.

Case 1: $j\neq 0$: Then $4^{\lfloor\log_4(n\dot{-}1)\rfloor}=4^{\lfloor\log_4 4k\rfloor}\leqslant 4k$. Hence $4p+3<4k$.

Case 2: $j=0$: Then $k\geqslant 2$ by $n=4k\geqslant 5$. Put $n-1=4k-1=\sum_{i=0}^{m}4^i\cdot y_i$ with $0\leqslant y_i<4$, $y_m\neq 0$ & $y_0=3$. Then $4^{\lfloor\log_4(n\dot{-}1)\rfloor}=4^m$. It suffices to show $4k-4-4^m\geqslant 0$ to have $4p+3<4k-4$. We have $4k-4-4^m=4^m\cdot(y_m-1)+\sum_{i<m}4^i\cdot y_i-3$. Suppose $m=0$. Then we would have $4\leqslant n-1=4k-1\leqslant 3$. A contradiction. Hence $m>0$. The assertion follows from $m>0$ and $y_0=3$.

Thus we have shown (15) and from this and $4p+3=p11$ we see

$$\begin{aligned}&Bit(p,or(and(x)))=1\leftrightarrow Bit(p0,and(x))=1\vee Bit(p1,and(x))=1\\&\quad\leftrightarrow(Bit(p00,x)=1\wedge Bit(p01,x)=1)\vee(Bit(p10,x)=1\wedge Bit(p11,x)=1).\end{aligned}$$

Thus we have $Tree(x,1p)\leftrightarrow Tree(or(and(x)),1p)$ as desired. □

Constructors. These are compositions and *CRN* (Concatenation Recursion on Notation): define f from g and $h_i\,(i<2)$ by

$$f(0,\bar{x})=g(\bar{x});\ \ n>0\rightarrow f(n,\bar{x})=s_i(f(\lfloor\tfrac{n}{2}\rfloor,\bar{x})),$$

where $i=sg(h_j(\lfloor n/2\rfloor,\bar{x}))$ with $j=parity(n)=Bit(0,n)$ and, $sg(0)=0$, $sg(x)=1$ for $x\neq 0$.

Composition is harmless and for *CRN* define

$$p_f(|n|,|\bar{x}|)=p_g|\bar{x}|+|n|$$

and

$$G_f(n,\bar{x},y)\leftrightarrow\exists|z|\leqslant p_g|\bar{x}|\left[g(\bar{x})=z\ \&\ \exists|u|\leqslant|n|\left\{y=u*z'\ \&\ \forall i<|n|\left(Bit(i,u)=0\leftrightarrow\bigvee_{j<2}\left(h_j\left(\left\lfloor\frac{n}{2}\right\rfloor,\bar{x}\right)=0\ \&\ Bit(i,n)=j\right)\right)\right\}\right],$$

where $z'=2^{|z|}+z$, $g(\bar{x})=z$ denotes $G_g(\bar{x},z)$, and $h_j(n,\bar{x})=i$ denotes $G_h(n,\bar{x},i)$. Note that by IH $G_h(n,\bar{x},i)$ is Δ_1^{b} in $AID+\Delta_1^{\mathrm{b}}$-CA and hence the RHS is Σ_1^{b}. Further a Π_1^{b}-form of $G_f(n,\bar{x},y)$ is defined similarly.

To show $\exists y G_f(n, \bar{x}, y)$ it suffices to show

$$\exists |u| \leqslant |n| \forall i < |n| \left(Bit(i,u)=0 \leftrightarrow \bigvee_{j<2} \left(h_j \left(\left\lfloor \frac{n}{2} \right\rfloor, \bar{x} \right) = 0 \ \& \ Bit(i,n)=j \right) \right)$$

and this follows from Δ_1^{b}-CA.

Thus we have shown Theorems 1 and 2.

9.2. Inductive definitions in ALV

In this subsection we show that *ALV* can simulate inductive definitions in *AID*. Specifically it is shown that for each inductively defined predicate A in *AID*, there exists a $\{0,1\}$-valued function symbol f_A in *ALV* such that $f_A(\bar{x}, p)=1$ satisfies the defining axioms (A.0)–(A.2) of the predicate A demonstrably in *ALV*.

Let *QALV* denote a quantified version of *ALV*. It is a first-order theory whose non-logical symbols are those of *ALV* and whose axioms are the universal closures of defining equations of these function symbols and induction on notation, together with two more axioms: $0 \neq 1$ and $\forall x(\lfloor x/2 \rfloor = 0 \rightarrow (x=0 \vee x=1))$. The last two axioms are added by Cook [16]. Note that in [16] the same name *QALV* designates a different first-order theory, i.e., a quantified version of ALV' in [12]. As in [16] we easily see:

Proposition 9.1. 1. *QALV is a conservative extension of ALV.*

2. *For each* Σ_0^{b}*-formula* B *in QALV there exists a function symbol* f_B *such that* $QALV \vdash B(\bar{x}) \leftrightarrow f_B(\bar{x})=0$, *and hence using CRN we have* $QALV \vdash \Sigma_0^{\mathrm{b}}$-CA: *for each* Σ_0^{b}*-formula* B *and each polynomial* p, $QALV \vdash \forall \bar{x} \exists y (y = \{i < p\,|\,\bar{x}\,| : B(\bar{x}, i)\})$.

3. *QALV proves* Σ_0^{b}-LIND.

Theorem 9.3. *For each inductively defined predicate* A *in AID, there exists a* $\{0,1\}$*-valued function symbol* f_A *in ALV such that* $f_A(\bar{x}, p)=1$ *satisfies the defining axioms* (A.0)–(A.2) *of the predicate* A *demonstrably in QALV.*

This together with Proposition 9.1 and Theorem 4.1 yields:

Corollary 9.1. *QALV and hence ALV prove reflection schema* $RFN(\mathscr{F})$ *for a Frege system.*

Now we prove Theorem 9.3. Let $A = A^{\ell, B, \bar{D}, I}$ be a given inductively defined predicate in *AID*. Recall that for a formula F and $i<2$ we have put

$$F^i = \begin{cases} F, & i=1, \\ \neg F, & i=0. \end{cases} \tag{2}$$

For $i<2$ put $k(i) = Bit(i,k)$.

First convert the boolean formulae I and $\neg I$ into DNF to yield: for $\xi < 2$,

$$\begin{aligned}
&I^{\xi}(\bar{D}(x,p), A(\bar{x},p0), A(\bar{x},p1)) \\
&\quad \leftrightarrow \bigvee \left\{ I^{\xi}_{k}(\bar{x},p) \wedge \bigwedge \{A(\bar{x},pi)^{k(i)}:\ i<2\}:\ k<2^{2} \right\} \\
&\quad \leftrightarrow \left[\bigvee \left\{ I^{\xi}_{k}(\bar{x},p) \wedge \bigwedge \{A(\bar{x},pi)^{k(i)}:\ i<2\}:\ k=0,1 \right\} \wedge (\top \vee \top) \right] \vee \\
&\quad\quad \left[\bigvee \left\{ I^{\xi}_{k}(\bar{x},p) \wedge \bigwedge \{A(\bar{x},pi)^{k(i)}:\ i<2\}:\ k=2,3 \right\} \wedge (\top \vee \top) \right]. \qquad (16)
\end{aligned}$$

Here is an *and/or* tree of depth 4. We define function symbols g^{ξ}_{A} so that for $\xi, \eta < 2$

$$tree(g^{\xi}_{A}(\bar{x},p)) = \eta \Leftrightarrow A^{\lambda}(\bar{x},p),$$

where $\lambda = (\xi \leftrightarrow \eta)$.

The resulting perfect *and/or* tree is of depth $4(\ell\|\bar{x}\| - |p|)$.

In view of (A.0), put

$$\neg(0 \neq |p| \leqslant \ell\|\bar{x}\|) \rightarrow g^{1}_{A}(\bar{x},p) = 0\ \&\ g^{0}_{A}(\bar{x},p) = 1.$$

In what follows suppose $0 \neq |p| \leqslant \ell\|\bar{x}\|$. We define function symbols h^{ξ}_{A} so that $g^{\xi}_{A}(\bar{x},p) = 2^{2^{4(\ell\|\bar{x}\| - |p|)}} + h^{\xi}_{A}(\bar{x},p)$ does the job. Put $y^{\xi} = h^{\xi}_{A}(\bar{x},p)$.

First consider the case $|p| = \ell\|\bar{x}\|$. Then by (A.1) we have $A(\bar{x},p) \leftrightarrow B(\bar{x},p)$. Put, cf. Proposition 9.1.2,

$$y^{\xi} = \begin{cases} 1 & \text{if } B^{\xi}(\bar{x},p), \\ 0 & \text{otherwise.} \end{cases}$$

Next, assume $|p| < \ell\|\bar{x}\|$. In the following we give a Σ^{b}_{0} condition in L_{BA} which is equivalent to $Bit(q, y^{\xi}) = 1$ for $q < |y^{\xi}| \leqslant 2^{4(\ell\|\bar{x}\| - |p|)}$ and hence $|q| \leqslant 4(\ell\|\bar{x}\| - |p|)$. Then by *CRN* we can pick a desired function symbol h^{ξ}_{A}.

Case 1: $\exists i \leqslant \lfloor |q|/4 \rfloor (Bit(4i+2, q) = 1)$: Put $Bit(q, y^{\xi}) = 1$. This corresponds to $(\top \vee \top)$ in (16). Namely, once the branch q has entered a definitely true subtree corresponding to $(\top \vee \top)$, the leaf of q receives 1.

Case 2: Suppose $\neg[\exists i \leqslant \lfloor |q|/4 \rfloor (Bit(4i+2, q) = 1)]$. For $m \leqslant \ell\|\bar{x}\| - |p|$ put

$$p_{m} = \{ i < m : Bit(4i, q[i_{1} - 4m, i_{1})) = 1 \}$$

with $i_{1} =_{\mathrm{df}} 4(\ell\|\bar{x}\| - |p|)$. Note that $q[i_{1} - 4m, i_{1})$ denotes the prefix of length $4m$ of the string $0^{[i_{1} - |q|]} * q$ over $\{0,1\}$ of length i_{1}.

Put $i_{0} =_{\mathrm{df}} i_{1} - 4m$. We say that q is *positive at* $m > 0$ if

$$(Bit(i_{0}, q) = 0\ \&\ Bit(i_{0}+1, q) = 1) \vee (Bit(i_{0}, q) = 1\ \&\ Bit(i_{0}+3, q) = 1).$$

This means that the subtree determined by the prefix $q[i_{0}, i_{1})$ evaluates the value of $A(\bar{x}, p \cdot 2^{m} + p_{m})$, cf. (16). Also we say that q is *positive at* $m = 0$ if $\xi = 1$. Otherwise q is said to be *negative at* $m \geqslant 0$.

Put $j = i_{0} - 4$ and let $k = Bit(j+3, q) Bit(j+1, q)$ denote $k < 4\ \&\ Bit(1,k) = Bit(j+3, q)\ \&\ Bit(0,k) = Bit(j+1, q)$ if $j = i_{0} - 4 = i_{1} - 4m - 4 \geqslant 0$, i.e., if $m < \ell\|\bar{x}\| -$

$|p|$. Otherwise, viz. in the case $m = \ell\|\bar{x}\| - |p|$, let $k = Bit(j+3,q)Bit(j+1,q)$ denote a $k<4$ and put $I_k^{\xi}(\bar{x}, p\cdot 2^m + p_m) \Leftrightarrow_{\mathrm{df}} B^{\xi}(\bar{x}, p\cdot 2^m + p_m)$ for any $k<4$.

Now the following is the necessary and sufficient condition to be $Bit(q, y^{\xi}) = 1$ in this *Case 2*:

$$\forall m \leqslant \ell\|\bar{x}\| - |\, p \,| \bigwedge_{k<4} [k = Bit(j+3,q)Bit(j+1,q)$$
$$\to \{q \text{ is positive at } m \to I_k^1(\bar{x}, p\cdot 2^m + p_m)\}\ \&$$
$$\{q \text{ is negative at } m \to I_k^0(\bar{x}, p\cdot 2^m + p_m)\}].$$

These give a Σ_0^{b}-definition of $Bit(q, y^{\xi}) = 1$. Then $\{(\bar{x}, p) : tree(g_A^{\xi}(\bar{x}, p)) = 1\}$ satisfies the defining axioms (A.0)–(A.2) for $A = A^{\ell, B, \bar{D}, I}$. The following Lemmata 9.1.2, 9.1.3 yield Theorem 9.3.

Lemma 9.1. 1. *Suppose* $z = 2^{2^{4m}} + y$ *with* $m>0$ *and* $y < 2^{2^{4m}}$. *For* $k<2^4$ *put* $z_k = 2^{2^{4(m-1)}} + y[2^{4(m-1)}k, 2^{4(m-1)}(k+1))$. *Let* $y_0 < 2^{2^4}$ *denote the number such that* $Bit(k, y_0) = tree(z_k)$ *for* $k<2^4$. *Then*

$$tree(z) = 1 \leftrightarrow or(and(or(and(2^{2^4} + y_0)))) = 1.$$

2. *Suppose* $0 \neq |p| < \ell\|\bar{x}\|$. *Then for* $i_1 = 4(\ell\|\bar{x}\| - |p|)$

$$tree(2^{2^{i_1}} + h_A^{\xi}(\bar{x}, p)) = 1$$
$$\leftrightarrow \bigvee \left\{ I_k^{\xi}(\bar{x}, p) \wedge \bigwedge \{tree(2^{2^{i_1-4}} + h_A^{k(i)}(\bar{x}, pi)) = 1 :\ i<2\} :\ k<2^2 \right\}.$$

3. *Suppose* $0 \neq |p| \leqslant \ell\|\bar{x}\|$. *Then*

$$tree(2^{2^{4(\ell\|x\|-|p|)}} + h_A^1(x, p)) = 1 \leftrightarrow tree(2^{2^{4(\ell\|x\|-|p|)}} + h_A^0(\bar{x}, p)) = 0.$$

Proof.
9.1.1. This is proved by induction on $m>0$.
9.1.2. Use Lemma 9.1.1.
9.1.3. This is proved by induction on $\ell\|\bar{x}\| - |p|$ using Lemma 9.1.2. □

9.3. Σ_1^{b}-*faithful interpretations*

In this subsection we conclude that translations derived from the preceding subsections are Σ_1^{b}-faithful.

For formula B in *QALV* let $I_{ID}(B)$ denote the formula in *AID* which is obtained from B by replacing each $f \in N_0$ by its Δ_1^{b}-graph defined in the proof of Theorems 9.1 and 9.2. Observe that for Σ_1^{b}-formula B in *QALV* $I_{ID}(B)$ is a Σ_1^{b}-formula in *AID*.

For a formula B in *AID* let $I_V(B)$ denote the formula in *QALV* which is obtained from B by replacing each $A^{\ell, B, \bar{D}, I}(\bar{x}, p)$ by a $tree(g_A^{\xi}(\bar{x}, p)) = 1$ defined in the proof of Theorems 9.3. Observe that for Σ^b_1-formula B in *AID* $I_V(B)$ is a Σ^b_1-formula in *QALV*.

Theorem 9.4. 1. *For each Σ^b_1-formula B in QALV,*

$$QALV \vdash B \iff AID + \Sigma^b_0\text{-CA} \vdash I_{ID}(B).$$

2. *For each Σ^b_1-formula B in AID,*

$$AID + \Sigma^b_0\text{-CA} \vdash B \iff QALV \vdash I_V(B).$$

Proof. Let T denote temporarily a union of theories $AID + \Sigma^b_0$-CA and $QALV$: its language L_T is the union $L_{AID} \cup L_{QALV}$, and its axioms are the ones in $AID + \Sigma^b_0(L_{AID})$-CA and $QALV$ *plus* $\Sigma^b_0(L_T)$-LIND. By Theorems 9.2 and 9.3 we have

1. For Σ^b_1-formula B in *QALV*, $T + \Delta^b_1(L_{AID})\text{-CA} \vdash B \leftrightarrow I_{ID}(B)$.
2. For Σ^b_1-formula B in *AID*, $T \vdash B \leftrightarrow I_V(B)$.

9.4.1. For a Σ^b_1-formula B in $QALV$ first suppose $QALV \vdash B$. Then $T + \Delta^b_1(L_{AID})\text{-CA} \vdash I_{ID}(B)$. By replacing any formula C in this proof by $I_{ID}(C)$ (leave formulae in *AID* unchanged) we get $AID + \Delta^b_1(L_{AID})\text{-CA} \vdash I_{ID}(B)$. Thus by Theorem 8.1.1 we conclude $AID + \Sigma^b_0\text{-CA} \vdash I_{ID}(B)$.

Conversely assume $AID + \Sigma^b_0\text{-CA} \vdash I_{ID}(B)$. Then $T + \Delta^b_1(L_{AID})\text{-CA} \vdash B$. By replacing any formula C in this proof by $I_V(C)$ (leave formulae in $QALV$ unchanged) we get $QALV + \Delta^b_1(L_{QALV})\text{-CA} \vdash B$. As in the proof of Theorem 8.1.1 or cf. Theorem 6 in p. 79, [16], we see that $QALV + \Sigma^b_1(L_{QALV})$-AC and hence $QALV + \Delta^b_1(L_{QALV})$-CA is Σ^b_1-conservative over $QALV$. Thus we conclude $QALV \vdash B$.

9.4.2. This is seen as in Theorem 9.4.1 using Proposition 9.1. □

Acknowledgements

I would like to thank S.A. Cook for inviting me to Fields Institute, Toronto hospitality during my stay and encouraging me to complete the work on *AID*. Without his interest and encouragement it would be impossible to finish this paper. I am very grateful to the anonymous referees for helpful suggestions and comments.

References

[1] M. Ajtai, The complexity of the pigeonhole principle, in: Proc. 29th Ann. IEEE Symp. on Foundations of Computer Science, 1988, pp. 346–355.

[2] J.L. Balcázar, J. Díaz, J. Gabarró, Structural Complexity II, EATCS Monographs on Theoret. Comput. Sci., Vol. 22, Springer, Berlin, 1988.

[3] M. Bonet, S. Buss, T. Pitassi, Are there hard examples for frege systems? in: P. Clote, J.B. Remmel (Eds.), Feasible Mathematics, Vol. II, Birkhäuser, Boston, 1995, pp. 30–56.

[4] S.R. Buss, Bounded Arithmetic, Bibliopolis, Napoli, 1986.

[5] S.R. Buss, Polynomial size proofs of the propositional pigeonhole principles, J. Symb. Logic 52 (1987) 916–927.

[6] S.R. Buss, The Boolean formula value problem is in *ALOGTIME*, in: Proc. 19 Ann. ACM Symp. on Theory of Computing, May 1987, pp. 123–131.
[7] S.R. Buss, Propositional consistency proofs, Ann. Pure. Appl. Logic 52 (1991) 3–29.
[8] S.R. Buss, The graph of multiplication is equivalent to counting, Inform. Proc. Lett. 14 (1992) 199–201.
[9] S.R. Buss, Algorithms for Boolean formula evaluation and tree contraction, in: P. Clote, J. Krajíček (Eds.), Arithmetic, Proof Theory and Computational Cimplexity, Oxford UP, Oxford, 1993, pp. 96–115.
[10] P. Clote, Sequential, machine-independent characterizations of the parallel complexity classes *ALOGTIME*, *ACk*, *NCk* and *NC*, in: S.R. Buss, P. Scott (Eds.), Feasible Mathematics, Birkhäuser, Boston Basel Berlin, 1990, pp. 49–70.
[11] P. Clote, *ALOGTIME* and a conjecture of S.A. Cook, Ann. Math. Art. Intell. 6 (1992) 57–106. extended abstract in Proc. IEEE Logic in Computer Science, Philadelphia, June, 1990.
[12] P. Clote, On polynomial size Frege proofs of certain combinatorial pinciples, in: P. Clote, J. Krajíček (Eds.), Arithmetic, Proof Theory and Computational Complexity, Oxford UP, Oxford, 1993, pp. 162–184.
[13] P. Clote, G. Takeuti, Bounded Arithmetics for *NC*, *ALOGTIME*, *L* and *NL*, Ann. Pure Appl. Logic 56 (1992) 73–117.
[14] P. Clote, G. Takeuti, First order bounded arithmetic and small boolean circuit complexity classes, in: P. Clote, J.B. Remmel (Eds.), Feasible Mathematics II, Birkhäuser, Boston Basel Berlin, 1995, pp. 154–218.
[15] S.A. Cook, Feasibly constructive proofs and the propositional calculus, in: Proc. 7th Ann. ACM Symp. on Theory of Computing, 1975, pp. 83–97.
[16] S.A. Cook, Relating the provable collapse of P to NC^1 and the power of logical theories, in: P. Beame, S. Buss (Eds.), Proof Complexity and Feasible Arithmetics, DIMACS series in Discrete Mathematics and Theoretical Computer Science Vol. 39, AMS, Providence, RI, 1998, pp. 73–91.
[17] J. Krajíček, On Frege and extended Frege proof systems, in: P. Clote, J.B. Remmel (Eds.), Feasible Mathematics II, Birkhäuser, Boston, 1995, pp. 284–319.
[18] J. Krajíček, Bounded Arithmetic, Propositional Logic, and Complexity Theory, Cambridge UP, Cambridge, 1995.
[19] F. Pitt, A quantifier-free theory based on a string algebra for NC^1, in: P. Beame, S. Buss (Eds.), Proof Complexity and Feasible Arithmetics, DIMACS series in Discrete Mathematics and Theoretical Computer Science, Vol. 39, AMS, Proidence, RI, 1998, pp. 229–252.